UNIVERSITE DES SCIENCES ET TECHNIQUES
DE LILLE FLANDRES ARTOIS

FLANDRES ARTOIS

N° d'ordre :

UNIVERSITE DES SCIENCES ET TECHNIQUES DE LILLE FLANDRES ARTOIS

THESE

présentée à

L'UNIVERSITE DES SCIENCES ET TECHNIQUES DE LILLE FLANDRES ARTOIS

pour obtenir le grade de

DOCTEUR D'ETAT ES SCIENCES MATHEMATIQUES

par

Florent CORDELLIER

INTERPOLATION RATIONNELLE ET AUTRES QUESTIONS : ASPECTS ALGORITHMIQUES ET NUMERIQUES.

Thèse soutenue le 19 juin 1989 devant la commission d'examen

Membres du jury

Président et rapporteur :	**J. VIGNES**
Rapporteurs :	**C. BREZINSKI** **P.R. GRAVES-MORRIS**
Examinateurs :	**J.C. FIOROT** **J. GILEWICZ** **L. WUYTACK**

Université des Sciences et Techniques de Lille Flandres-Artois
Laboratoire d'Analyse Numérique et d'Optimisation - Bât M3 - 1e étage
U.F.R. d'I.E.E.A.
59655 VILLENEUVE D'ASCQ - CEDEX - FRANCE
Tél. 20.43.46.71

UNIVERSITE DES SCIENCES
ET TECHNIQUES DE LILLE
FLANDRES ARTOIS

DOYENS HONORAIRES DE L'ANCIENNE FACULTE DES SCIENCES

M.H. LEFEBVRE, M. PARREAU.

PROFESSEURS HONORAIRES DES ANCIENNES FACULTES DE DROIT ET SCIENCES ECONOMIQUES, DES SCIENCES ET DES LETTRES

MM. ARNOULT, BONTE, BROCHARD, CHAPPELON, CHAUDRON, CORDONNIER, DECUYPER, DEHEUVELS, DEHORS, DION, FAUVEL, FLEURY, GERMAIN, GLACET, GONTIER, KOURGANOFF, LAMOTTE, LASSERRE, LELONG, LHOMME, LIEBAERT, MARTINOT-LAGARDE, MAZET, MICHEL, PEREZ, ROIG, ROSEAU, ROUELLE, SCHILTZ, SAVARD, ZAMANSKI, Mes BEAUJEU, LELONG.

PROFESSEUR EMERITE

M. A. LEBRUN

ANCIENS PRESIDENTS DE L'UNIVERSITE DES SCIENCES ET TECHNIQUES DE LILLE

MM. M. PAREAU, J. LOMBARD, M. MIGEON, J. CORTOIS.

PRESIDENT DE L'UNIVERSITE DES SCIENCES ET TECHNIQUES DE LILLE FLANDRES ARTOIS

M. A. DUBRULLE.

PROFESSEURS - CLASSE EXCEPTIONNELLE

M. CONSTANT Eugène	Electronique
M. FOURET René	Physique du solide
M. GABILLARD Robert	Electronique
M. MONTREUIL Jean	Biochimie
M. PARREAU Michel	Analyse
M. TRIDOT Gabriel	Chimie Appliquée

PROFESSEURS - 1ère CLASSE

M. BACCHUS Pierre	Astronomie
M. BIAYS Pierre	Géographie
M. BILLARD Jean	Physique du Solide
M. BOILLY Bénoni	Biologie
M. BONNELLE Jean-Pierre	Chimie-Physique
M. BOSCQ Denis	Probabilitéss
M. BOUGHON Pierre	Algèbre
M. BOURIQUET Robert	Biologie Végétale
M. BREZINSKI Claude	Analyse Numérique

M. BRIDOUX Michel	Chimie-Physique
M. CELET Paul	Géologie Générale
M. CHAMLEY Hervé	Géotechnique
M. COEURE Gérard	Analyse
M. CORDONNIER Vincent	Informatique
M. DAUCHET Max	Informatique
M. DEBOURSE Jean-Pierre	Gestion des Entreprises
M. DHAINAUT André	Biologie Animale
M. DOUKHAN Jean-Claude	Physique du Solide
M. DYMENT Arthur	Mécanique
M. ESCAIG Bertrand	Physique du Solide
M. FAURE Robert	Mécanique
M. FOCT Jacques	Métallurgie
M. FRONTIER Serge	Ecologie Numérique
M. GRANELLE Jean-Jacques	Sciences Economiques
M. GRUSON Laurent	Algèbre
M. GUILLAUME Jean	Microbiologie
M. HECTOR Joseph	Géométrie
M. LABLACHE-COMBIER Alain	Chimie Organique
M. LACOSTE Louis	Biologie Végétale
M. LAVEINE Jean-Pierre	Paléontologie
M. LEHMANN Daniel	Géométrie
Mme LENOBLE Jacqueline	Physique Atomique et Moléculaire
M. LEROY Jean-Marie	Spectrochimie
M. LHOMME Jean	Chimie Organique Biologique
M. LOMBARD Jacques	Sociologie
M. LOUCHEUX Claude	Chimie Physique
M. LUCQUIN Michel	Chimie Physique
M. MACKE Bruno	Physique Moléculaire et Rayonnements Atmosph.
M. MIGEON Michel	E.U.D.I.L.
M. PAQUET Jacques	Géologie Générale
M. PETIT Francis	Chimie Organique
M. POUZET Pierre	Modélisation - calcul Scientifique
M. PROUVOST Jean	Minéralogie
M. RACZY Ladislas	Electronique
M. SALMER Georges	Electronique
M. SCHAMPS Joel	Spectroscopie Moléculaire
M. SEGUIER Guy	Electrotechnique
M. SIMON Michel	Sociologie
Melle SPIK Geneviève	Biochimie
M. STANKIEWICZ François	Sciences Economiques
M. TILLIEU Jacques	Physique Théorique
M. TOULOTTE Jean-Marc	Automatique
M. VIDAL Pierre	Automatique
M. ZEYTOUNIAN Radyadour	Mécanique

PROFESSEURS - 2ème CLASSE

M. ALLAMANDO Etienne	Composants Electroniques
M. ANDRIES Jean-Claude	Biologie des organismes
M. ANTOINE Philippe	Analyse
M. BART André	Biologie animale
M. BASSERY Louis	Génie des Procédés et Réactions Chimiques

Mme BATTIAU Yvonne	Géographie
M. BEGUIN Paul	Mécanique
M. BELLET Jean	Physique Atomique et Moléculaire
M. BERTRAND Hugues	Sciences Economiques et Sociales
M. BERZIN Robert	Analyse
M.BKOUCHE Rudolphe	Algèbre
M. BODARD Marcel	Biologie Végétale
M. BOIS Pierre	Mécanique
M. BOISSIER Daniel	Génie Civil
M. BOIVIN Jean-Claude	Spectroscopie
M. BOUQUELET Stéphane	Biologie Appliquée aux enzymes
M. BOUQUIN Henri	Gestion
M. BRASSELET Jean-Paul	Géométrie et Topologie
M. BRUYELLE Pierre	Géographie
M. CAPURON Alfred	Biologie Animale
M. CATTEAU Jean-Pierre	Chimie Organique
M. CAYATTE Jean-Louis	Sciences Economiques
M. CHAPOTON Alain	Electronique
M. CHARET Pierre	Biochimie Structurale
M. CHIVE Maurice	Composants Electroniques Optiques
M. COMYN Gérard	Informatique Théorique
M. COQUERY Jean-Marie	Psychophysiologie
M. CORIAT Benjamin	Sciences Economiques et Sociales
Mme CORSIN Paule	Paléontologie
M. CORTOIS Jean	Physique Nucléaire et Corpusculaire
M. COUTURIER Daniel	Chimie Organique
M. CRAMPON Norbert	Tectolique Géodynamique
M. CROSNIER Yves	Electronique
M. CURGY Jean-Jacques	Biologie
Melle DACHARRY Monique	Géographie
M. DEBRABANT Pierre	Géologie Appliquée
M. DEGAUQUE Pierre	Electronique
M. DEJAEGER Roger	Electrochimie et Cinétique
M. DELAHAYE Jean-Paul	Informatique
M. DELORME Pierre	Physioliogie Animale
M. DELORME Robert	Sciences Economiques
M. DEMUNTER Paul	Sociologie
M. DENEL Jacques	Informatique
M. DE PARIS Jean Claude	Analyse
M. DEPREZ Gilbert	Physique du Solide - Cristallographie
M. DERIEUX Jean-Claude	Microbiologie
Melle DESSAUX Odile	Spectroscopie de la réactivité Chimique
M. DEVRAINNE Pierre	Chimie Minérale
Mme DHAINAUT Nicole	Biologie Animale
M. DHAMELINCOURT Paul	Chimie Physique
M. DORMARD Serge	Sciences Economiques
M. DUBOIS Henri	Spectroscopie Hertzienne
M. DUBRULLE Alain	Spectroscopie Hertzienne
M. DUBUS Jean-Paul	Spectrométrie des Solides
M. DUPONT Christophe	Vie de la firme (I.A.E.)
Mme EVRARD Micheline	Génie des procédés et réactions chimiques
M. FAKIR Sabah	Algèbre
M. FAUQUAMBERGUE Renaud	Composants électroniques

M. FONTAINE Hubert	Dynamique des cristaux
M. FOUQUART Yves	Optique atmosphérique
M. FOURNET Bernard	Biochimie Sturcturale
M. GAMBLIN André	Géographie urbaine, industrielle et démog.
M. GLORIEUX Pierre	Physique moléculaire et rayonnements Atmos.
M. GOBLOT Rémi	Algèbre
M. GOSSELIN Gabriel	Sociologie
M. GOUDMAND Pierre	Chimie Physique
M. GOURIEROUX Christian	Probabilités et Statistiques
M. GREGORY Pierre	I.A.E.
M. GREMY Jean-Paul	Sociologie
M. GREVET Patrice	Sciences Economiques
M. GRIMBLOT Jean	Chimie Organique
M. GUILBAULT Pierre	Physiologie animale
M. HENRY Jean-Pierre	Génie Mécanique
M. HERMAN Maurice	Physique spatiale
M. HOUDART René	Physique atomique
M. JACOB Gérard	Informatique
M. JACOB Pierre	Probabilités et Statistiques
M. Jean Raymond	Biologie des populations végétales
M. JOFFRE Patrick	Vie de la firme (I.A.E.)
M. JOURNEL Gérard	Spectroscopie hertzienne
M. KREMBEL Jean	Biochimie
M. LANGRAND Claude	Probabilités et statistiques
M. LATTEUX Michel	Informatique
Mme LECLERCQ Ginette	Catalyse
M. LEFEBVRE Jacques	Physique
M. LEFEBVRE Christian	Pétrologie
Melle LEGRAND Denise	Algèbre
Melle LEGRAND Solange	Algèbre
M. LEGRAND Pierre	Chimie
Mme LEHMANN Josiane	Analyse
M. LEMAIRE Jean	Spectroscopie hertzienne
M. LE MAROIS Henri	Vie de la firme (I.A.E.)
M. LEROY Yves	Composants électroniques
M. LESENNE Jacques	Systèmes électroniques
M. LHENAFF René	Géographie
M. LOCQUENEUX Robert	Physique théorique
M. LOSFELD Joseph	Informatique
M. LOUAGE Francis	Electronique
M. MAHIEU Jean-Marie	Optique-Physique atomique
M. MAIZIERES Christian	Automatique
M. MAURISSON Patrick	Sciences Economiques et Sociales
M. MESMACQUE Gérard	Génie Mécanique
M. MESSELYN Jean	Physique atomique et moléculaire
M. MONTEL Marc	Physique du solide
M. MORCELLET Michel	Chimie Organique
M. MORTREUX André	Chimie Organique
Mme MOUNIER Yvonne	Physiologie des structures contractiles
Mme MOUYART-TASSIN Annie Françoise	Informatique
M. NICOLE Jacques	Spectrochimie
M. NOTELET Françis	Systèmes électroniques
M. PARSY Fernand	Mécanique

M. PECQUE Marcel	Chimie organique
M. PERROT Pierre	Chimie appliquée
M. STEEN Jean-Pierre	Informatique

Je suis très honoré par la présence de M. Jean Vignes, Professeur à l'Université de Paris VI, qui a accepté de présider ce Jury.

Je tiens à exprimer toute ma gratitude à M. Claude Brezinski, Professeur à l'Université des Sciences et Techniques de Lille-Flandres-Artois, qui n'a jamais cessé de m'encourager dans la bien longue préparation de ce travail.

Je souhaite que M. Peter Graves-Morris, Professeur à l'Université de Bradford, sache combien je lui suis obligé d'avoir accepté de se déplacer tout spécialement pour venir juger ce travail.

Je remercie M. Luc Wuytack, Professeur à l'Université d'Anvers, d'avoir accepté de participer au Jury.

Que M. Jean-Charles Fiorot, Professeur à l'Université de Valenciennes et du Hainaut-Cambrésis, et M. Jacek Gilewicz, Professeur à l'Université de Toulon trouvent ici le témoignage de ma reconnaissance pour avoir accepté de venir juger ce travail après m'avoir autant poussé à le concrétiser.

Ce travail a été réalisé au sein de l'équipe d'Analyse Numérique et d'Optimisation de l'Université de Lille I. Je tiens à remercier tous les membres de l'équipe pour leur collaboration et je souhaite en particulier saluer ici Bernard Germain-Bonne dont le concours m'a été si précieux. Je tiens également à saluer également MM. Bernard Sucher et Bernard Leguy, du laboratoire d'Informatique, avec lesquels j'ai eu si souvent des discussions enrichissantes.

Enfin, mes remerciements vont à Madame Françoise Tailly, qui a assuré la frappe de l'annexe 2, et à Monsieur Glanc, qui a assuré la présentation générale du travail.

INTERPOLATION RATIONNELLE ET AUTRES QUESTIONS: ASPECTS NUMÉRIQUES ET ALGORITHMIQUES

0. Introduction

Le travail que je présente aujourd'hui comporte deux types de documents: des papiers déjà publiés dont la liste figure dans le paragraphe 6 et des textes non publiés ou de diffusion trop restreinte qui constituent les annexes 1 à 6.

L'ensemble de ces textes peut être ventilé sous cinq rubriques: la première se rapporte à un problème d'optimisation auquel une collaboration fructueuse avec J. C. Fiorot m'a permis d'apporter une solution. Les deux suivantes concernent des thèmes privilégiés de l'équipe de recherche dirigée par C. Brezinski à laquelle j'ai le plaisir d'appartenir: l'étude des transformations élémentaires de suites s'inscrit dans le cadre de l'accélération de la convergence et l'interpolation rationnelle généralisée est étroitement liée à l'étude des approximants de Padé. Les deux dernières rubriques, plus personnelles, sont relatives à une question qui m'obsède depuis longtemps: traduire des algorithmes numériques dont la fiabilité risque d'être doûteuse en des programes fiables dont la stabilité numérique est garantie. La quatrième partie illustre cette démarche sur une classe de problèmes précis, la mise en œuvre des algorithmes de losange, tandis que la dernière rassemble des textes liés à la représentation des nombres et aux problèmes que pose cette représentation.

Bien entendu, la distinction précédente est plus apparente que réelle car les quatre dernières rubriques ne peuvent être aussi aisément dissociées: la mise en œuvre des transformations de suites repose le plus souvent sur des algorithmes d'interpolation rationnelle parmi lesquels les algorithmes de losange occupent une place de choix; enfin peut-on parler de la mise en œuvre de ces algorithmes de losange sans s'intéresser de près aux aspects numériques de la programmation?

Chacune de ces questions est présentée dans l'un des cinq paragraphes qui suivent. Dans le paragraphe 6, j'ai rassemblé la liste des travaux que je présente aujourd'hui regroupés autour des cinq thèmes retenus. Treize d'entre eux ont été diffusés sous la forme de publications dans des revues ou dans les actes de colloques internationaux tandis que six autres n'ont connu qu'une diffusion restreinte. Leur texte est adjoint au présent mémoire sous la forme d'annexes numérotées de 1 à 6. Enfin, on trouvera dans le paragraphe 7 la liste des travaux auxquels je me suis référé dans la rédaction de ce mémoire (y compris dans les papiers déjà diffusés qui ne figurent pas en annexe).

1. Le problème de Fermat-Weber généralisé.

1.2 *Position du problème.*

On se donne m points a_i de $\mathbb{R}^n$ à chacun desquels sont associées deux fonctions:

$d_i(x) = \| x - a_i \|$, distance euclidienne de x à a_i,

$f_i : \mathbb{R} \to \mathbb{R}$, fonction convexe, différentiable et non décroissante.

Le problème consiste à déterminer un point x satisfaisant:

$$(1) \quad \min \{ \sum_{i=1}^{m} f_i(d_i(x)) \mid x \in \mathbb{R}^n \}$$

Ce problème est une extension proposée par Katz [69] du classique problème de Fermat-Weber généralisé [127] qu'on retrouve en posant:

(2) $f_i(u) = w_i u$, où le poids w_i est fixé.

Il a de nombreuses applications en économie: on peut en particulier l'interpréter comme la localisation du centre d'un réseau dont les éléments sont les a_i. Il s'agit donc d'un problème d'optisation

sans contrainte dont la fonction économique est différentiable sauf aux points a_i. Cette non-différentiabilité de la fonction économique aux points a_i est la difficulté majeure du problème.

1.2 *Les solutions proposées.*

Pour résoudre ce problème, un algorithme itératif a été proposé par Weiszfeld [128] en 1937, mais cet algorithme est en défaut si un itéré est confondu avec un a_i distinct de l'optimum. Jacobsen [65] a proposé un itéré particulier dans le cas où une telle éventualité se produisait, et, dans un papier non publié, Jacobsen, Fiorot et moi-même avons établi la convergence du procédé.

Par la suite, Fiorot et moi-même avons proposé trois algorithmes itératifs distincts pour résoudre ce problème de Fermat-Weber généralisé (FW1), puis nous avons étendu deux d'entre eux à la résolution du problème modifié par Katz (FW2,FW3). L'idée de ces algorithmes consiste à majorer globalement la fonction économique non différentiable par une fontion économique différentiable dépendant du point d'itération, fonction dont on choisit le minimum comme itéré. La convergence de tous les algorithmes est établie quel que soit le point de départ.

2 Les transformations de suites.

2.1 *Introduction.*

Dans ce chapitre sont réunis trois papiers concernant des propriétés intrinsèques des transformations de suites, en excluant leurs aspects algorithmiques. Les deux premiers se rapportent à deux transformations élémentaires de suites, à savoir: une généralisation du procédé δ^2 d'Aitken [2] obtenue en introduisant un paramètre α pour définir le procédé γ_α et la transformation W de Lubkin [85]. Le troisième présente une généralisation de la transformation E_k de Shanks [117] basée sur la notion d'approximation.

2.2 *Caractérisation des noyaux de γ_α et W.*

Le premier papier (TS1) caractérise le noyau, c'est à dire l'ensemble des suites qu'un tel procédé transforme en une suite identiquement nulle. Grace à la quasi-linéarité de ces transformations, la connaissance de ce noyau permet caractériser les suites que ce procédé transforme en suites constantes. Le résultat essentiel est que le noyau de W contient la réunion des noyaux de γ_α quand α parcourt $\mathbb{C}$. Dans ce papier, nous avons supposé que les transformations ne s'appliquent pas à des suites dont deux termes consécutifs sont égaux. On peut s'affranchir de cette contrainte sans modifier le résultat essentiel au prix de démonstrations sensiblement plus complexes.

2.3 *Régularité,quasi-régularité et compatibilité des procédés γ_α et W.*

La motivation topologique de ce second papier (TS2) s'oppose à la motivation algébrique du premier. Trois questions y sont posées: la régularité, la quasi-régularité et la compatibilité des deux procédés.
On caractérise tout d'abord le domaine de régularité des transformations, c'est à dire l'ensemble des suites dont elles préservent la convergence. Ces transformations ne sont pas régulières car leur domaine de régularité ne couvre pas l'ensemble des suites convergentes.
A défaut de la régularité des transformations, on peut s'intéresser à leur quasi-régularité, c'est à dire au fait que , lorsqu'elles conservent la convergence d'une suite, elles ne modifient pas sa limite. Pour le procédé δ^2, ce résultat avait été établi par Lubkin [85] dans le cas des suites de réels et par Tucker [123] pour les suites de complexes. On l'étend ici à l'ensembles des procédés γ_α pour les suites de complexes et on l'établit dans le cas des suites de réels pour le procédé W.

Nous dirons qu'une suite x est T-limitable si sa transformée par un procédé T converge vers une limite qu'on appellera T-limite de la suite x. Si la suite x converge, l'identité de la limite et de la T-limite est garantie par la quasi-régularité de la transformation T. L'intérêt de cette notion provient

du fait que la suite x peut diverger et avoir une T-limite (Euler [61]). Pour qu'une telle T-limite présente un intérêt, il faut encore qu'elle ne dépende pas de façon trop fantaisiste du choix de la transformation. On montre que, si une suite de nombres réels est simultanément γ_α-limitable et W-limitable, alors sa γ_α-limite et sa W-limite sont égales. A notre connaissance, ce résultat est le seul de cette nature dans le domaine des transformations non linéaires de suites.

2.4 *Une généralisation de la transformation de Shanks.*

2.4.1 Présentation de la nouvelle transformation.

La transformation $E_k(s_n)$ de Shanks [117], que l'ε-algorithme de Wynn [141] permet de mettre en œuvre très commodément, repose sur l'idée d'interpolation: une suite (s_n) étant donnée, $E_k(s_n)$ est la limite (ou l'anti-limite [117]) d'une suite t qui vérifie les deux conditions suivantes:

1) elle interpole la suite s aux 2k+1 indices: n,n+1,n+2, ...,n+2k;
2) elle satisfait une récurrence linéaire d'ordre k.

L'idée de la transformation proposée ici consiste à renforcer la première contrainte en relaxant la seconde pour obtenir:

une suite (s_n) étant donnée, pour 2 entiers k et m donnés vérifiant $0<2k\leq m$, $C_k^m(s_n)$ est la limite (ou l'anti-limite) d'une suite t satisfaisant les deux contraintes suivantes:

1) t interpole la suite s aux m+1 indices : n,n+1,n+2,...,n+m;
2) t satisfait *au mieux* une récurrence linéaire d'ordre k.

Dans le cas où $m=2k$, $C_k^{2k}(s_n)$ s'identifie à $E_k(s_n)$. Dans le cas général (où $m\geq 2k$), cette transformation s'exprime encore au moyen d'un quotient de déterminants, et les algorithmes MNA ou RPA proposés par Brezinski [8] permettent de la mettre en œuvre récursivement.

2.4.2 Propriétés des transformations $C_1^m(s_n)$.

Une étude plus approfondie de la plus simple de ces transformations, le procédé $C_1^m(s_n)$ qui généralise le procédé δ^2 d'Aitken [117], permet de montrer que:

1) C_1^m est quasi-régulière sur les suites de réels;

2) Les suites à convergence *logarithmique* sont *accélérées* par ce procédé;

3) Le *conditionnement* du procédé C_1^m est d'autant meilleur que m est grand. Une exprimentation numérique vient confirmer ce dernier résultat. Nous renvoyons à la lecture du papier pour voir le sens qu'on peut prêter aux mots en italique.

3. L'interpolation rationnelle généralisée.

Bien que ne proposant pas explicitement d'algorithme, la motivation de ces deux papiers est essentiellement algorithmique.

Dans le premier (IR1), un formalisme général permettant de couvrir une vaste classe de problèmes d'interpolation rationnelle est proposé. La solution des problèmes posés est présentée sous la forme de quotients de déterminants, et le calcul de ces quotients est très brièvement esquissé.

Le second (IR2) se rapporte encore à l'interpolation rationnelle généralisée, mais l'objectif est ici très spécifique: il s'agit de montrer que l'algorithme de Kronecker [72] peut être adapté à ce type de problème pour fournir un algorithme totalement fiable. On sait que, dans des conditions d'utilisation convenables, cet algorithme fournit tous les interpolants rationnels dont la somme des degrés du numérateur et du dénominateur est fixée. La définition de cet algorithme repose sur la notion de

quotient de formes rationnelles introduite dans le papier. Le fait que tous les interpolants sont obtenus au moyen de cet algorithme est établi en appliquant (de façon théorique) une forme duale de l'algorithme présenté.

4. Fiabilité et stabilité des algorithmes de losange.

4.1 *Introduction et motivation.*

Les algorithmes de losange interviennent couramment en interpolation rationnelle: algorithme de Thiele [122] et de Claessens [29], variante de l'algorithme de Stoer-Larkin [81,120] ou valeurs ponctuelles d'une table de Padé [140]. Ils interviennent encore dans la mise en œuvre de certaines transformations de suites: ρ et ε-algorithmes scalaires [141,142], vectoriels [139] et leurs généralisations [11,14] . Ils consistent à construire une suite de suites (θ_i) à partir d'une initialisation fixant les deux premières suites et à calculer les termes des suites suivantes en utilisant une relation de récurrence:

Initialisation: $\theta_{-1}^j = v_j$, $\theta_0^j = x_j$, $j \in \mathbb{N}$.

Récurrence: $\theta_{i+1}^j = \theta_{i-1}^{j+1} + a_i^j / (\theta_i^{j+1} - \theta_i^j)$, $i,j \in \mathbb{N}$.

où les coefficients a_i^j et les suites initiales (v_j) et (x_j) caractérisent l'algorithme.

Dans la notation θ_i^j, l'indice inférieur i identifie une suite tandis que l'indice supérieur j représente l'indice du terme dans la suite θ_i. Les éléments θ_i^j sont placés dans un tableau à deux indices où l'indice inférieur i repère une colonne et l'indice supérieur j une diagonale descendante (figure 1).

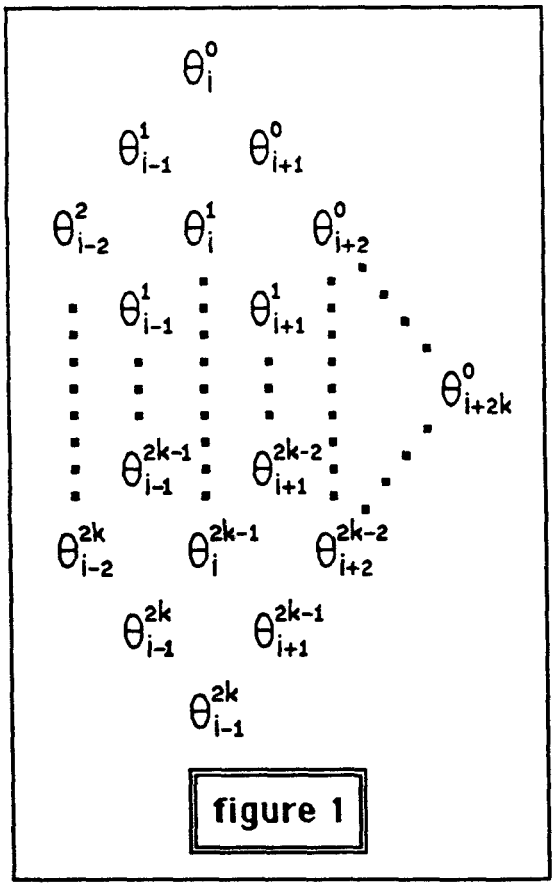


figure 1

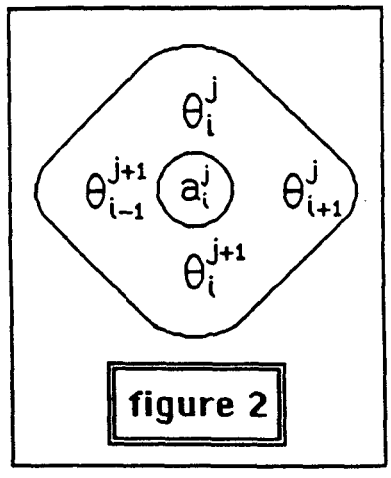


figure 2

Chaque relation de récurrence fait donc intervenir les éléments situés aux quatre sommets d'un losange et dépend d'un coefficient a_i^j attaché à ce losange (figure 2).

La mise en œuvre de ces algorithmes est handicapée par deux difficultés:
1) leur fiabilité est doûteuse: l'égalité de deux éléments consécutifs d'une même colonne interdit le calcul de certains éléments de la colonne suivante;
2) leur stabilité numérique est médiocre: la trop grande proximité de deux éléments consécutifs d'une même colonne induit une grande erreur de calcul dans certains éléments de la colonne suivante.
Bien entendu, tout problème rencontré lors de la construction d'une colonne est hérité et même amplifié dans les colonnes suivantes.

4.2 *Solutions apportées.*

Dès 1963, Wynn [143] a proposé deux règles pour lutter contre la non-fiabilité et l'instabilité numérique de l'ε-algorithme scalaire lorsque deux valeurs consécutives d'une même colonne sont égales (non fiabilité) ou voisines (instabilité). Brezinski [15] a présenté une règle similaire pour le ρ-algorithme scalaire, et j'ai également proposé une règle valable dans le cas de l'ε-algorithme vectoriel (FS4). Toutefois ,le problème n'est pas règlé pour autant: que faire si plus de deux valeurs consécutive d'une même colonne sont égales ou même simplement voisines? L'extension des règles précédentes s'avère nécessaire.

La solution que je propose repose sur une notion fondamentale: l'invariance anallagmatique des colonnes de même parité de tout algorithme de losange dont les coefficients vérifient une certaine propriété. Esquissons ici brièvement les grandes lignes de la démarche qui aboutit aux résultats suivants:
1) extension de l'identité de Wynn [144] dans le cas vectoriel,d'où une version fiable (mais non nécessairement numériquement stable) de l'ε-algorithme vectoriel;
2) versions fiables et stables de tous les algorithmes de losange dont les coefficients vérifient une certaine propriété.
Sans hypothèse complémentaire, les éléments θ_i^j sont liés par l'identité suivante que nous appellerons *règle de la croix*, valable pour tout couple (i,j) d'entiers non négatifs:

$$a_i^j(\theta_i^j-\theta_i^{j+1})^{-1}+a_i^{j+1}(\theta_i^{j+2}-\theta_i^{j+1})^{-1}=a_{i+1}^j(\theta_{i+2}^j-\theta_i^{j+1})^{-1}+a_{i-1}^{j+1}(\theta_{i-2}^{j+2}-\theta_i^{j+1})^{-1}$$

A tout algorithme de losange, on peut donc associer deux algorithmes qui ne font intervenir que les colonnes dont l'indice a la même parité, chacun étant initialisé au moyen de deux colonnes: les colonnes d'indices -2 et 0 pour l'un, celles d'indices -1 et +1 pour l'autre. Nous nous réfèrerons à ces algorithmes comme à la forme en croix (paire ou impaire) d'un algorithme de losange. Notons encore que, si l'on connait toutes les colonnes dont l'indice est pair (respectivement impair), il suffit d'une valeur supplémentaire dans l'une quelconque des autres colonnes pour définir les valeurs des colonnes ayant l'autre parité.

Rappelons qu'on qualifie d'anallagmatique toute transformation ponctuelle qui conserve la nature des cercles et des sphères (en considérant les hyperplans comme des hypersphères contenant le point à l'infini) et le birapport de quatre points sur un cercle. Ces transformations constituent le sous-groupe des transformations ponctuelles de l'espace engendré par les similitudes et les inversions. Toute la suite repose sur la remarque suivante:
Pourvu que les poids a_i^j satisfassent la relation:

$$a_i^j+a_i^{j+1}=a_{i-1}^{j+1}+a_{i+1}^j,\quad \forall\ i,j\in\mathbb{N},$$

la règle de la croix est anallagmatiquement invariante.

Nous nous restreindrons désormais aux algorithmes de losange dont les poids satisfont cette propiété. Appliquons une transformation anallagmatique A à deux colonnes consécutives de même parité d'un tableau T construit lors de la mise en œuvre d'un algorithme de losange.

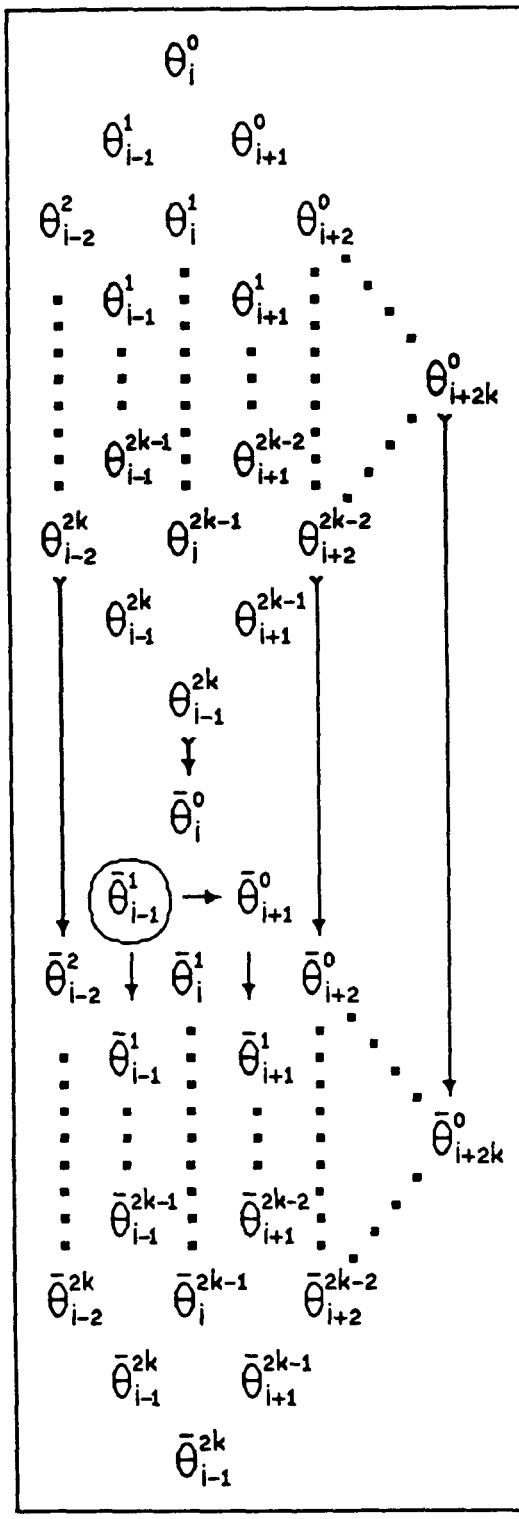


Tableau initial

Tableau transformé

dont l'initialisation est définie par:

$$\begin{cases} \bar{\theta}_i^j = A(\theta_i^j),\ \forall j \geqslant 0. \\ \theta_{i-2}^j = A(\theta_{i-2}^j),\ \forall j \geqslant 2. \\ \text{et} \\ \bar{\theta}_{i-1}^1 \text{ fixé arbitrairement.} \end{cases}$$

figure 3

A partir de l'initialisation définie par les transformées par A des deux colonnes en question, mettons en œuvre la forme en croix de l'algorithme de losange utilisant les mêmes coefficients que ceux utilisés lors de la construction de T. Alors les colonnes de même parité que l'initialisation construites au moyen de cette technique sont les transformées par A des colonnes de même indice du tableau initial (figure 3).

Dans toute la suite, nous n'utiliserons qu'une seule transformation anallagmatique: l'inversion ayant pour pôle l'origine et pour puissance l'unité qui offre l'avantage d'être numériquement la plus stable de ces transformations. La propriété d'invariance anallagmatique peut alors être utilisée de deux façons distinctes que nous qualifierons respectivement de systématique et sélective.

4.2.2 Utilisation systématique

Notons ${}^{0}\theta_i^j$ les éléments calculés au moyen de l'algorithme de losange initial. Dès que la colonne d'indice 1 est connue, on peut inverser les éléments des deux colonnes d'indices respectifs -1 et 1 qu'on retient comme initialisation d'un nouvel algorithme de losange utilisant les mêmes coefficients et dont les éléments seront notés ${}^{1}\theta_i^j$. Pour assurer une mise en œuvre commode de cet algorithme, on fixera arbitrairement une valeur dans les colonnes d'indice pair, par exemple ${}^{1}\theta_0^1 = 0$, ce qui permet de calculer toutes les colonnes du nouveau tableau.

On itère le procédé en inversant les éléments des colonnes d'indices 0 et 2 du tableau ${}^{1}\theta_i^j$ pour des éléments qu'on choisit comme initialisation d'un nouvel algorithme de losange utilisant encore les mêmes coefficients et dont les éléments seront notés ${}^{2}\theta_i^j$. On introduira des colonnes impaires dans ce tableau en fixant arbitrairement une valeur, par exemple ${}^{2}\theta_i^0 = 0$.

On pourra ainsi définir des tableaux successifs $T^{(0)}, T^{(1)}, T^{(2)}, \ldots, T^{(k)}, \ldots$, dont les éléments sont notés ${}^{0}\theta_i^j, {}^{1}\theta_i^j, {}^{2}\theta_i^j, \ldots, {}^{k}\theta_i^j, \ldots$.

Les éléments homologues ${}^{m-1}\theta_i^j$ et ${}^{m}\theta_i^j$ de deux tableaux consécutifs $T^{(m-1)}$ et $T^{(m)}$ sont inverses l'une de l'autre s'ils appartiennent à des colonnes de même parité, donc si les indices m et i ont même parité. Si nous disposons d'un moyen de connaître la précision avec laquelle est connu un élément calculé, on peut alors comparer les éléments homologues de deux tableaux consécutifs, retenir celui qui est le plus précis et recalculer l'autre par inversion. Cela reste valable si la précision d'un des deux nombres est nulle, c'est à dire si ce nombre est indéterminé, ce qui permet d'améliorer la fiabilité.
L'utilisation systématique de cette idée conduit à la version fiable et stable de divers algorithmes de losange proposée en (FS6),version dans laquelle l'estimation de l'erreur est fournie par la méthode de perturbation aléatoire de Vignes [79,80,125]. Bien entendu, pour des raisons d'économie, on peut limiter le nombre des tableaux créés, donc le coût de l'algorithme.

4.2.3 Utilisation sélective.

On peut aussi circonscrire la mise en œuvre de cette technique d'inversion exploitant l'invariance anallagmatique d'un algorithme de losange à des zônes bien délimitées du tableau construit, zônes dans lesquelles on aura détecté l'instabilité numérique ou même la non-fiabilité de l'algorithme initial. Cette méthode sélective est simplifiée si l'on dispose d'informations précises sur la structure des singularités, ce qui est précisément le cas le l'ε-algorithme (scalaire ou vectoriel). On peut alors l'exploiter dans deux contextes distincts:

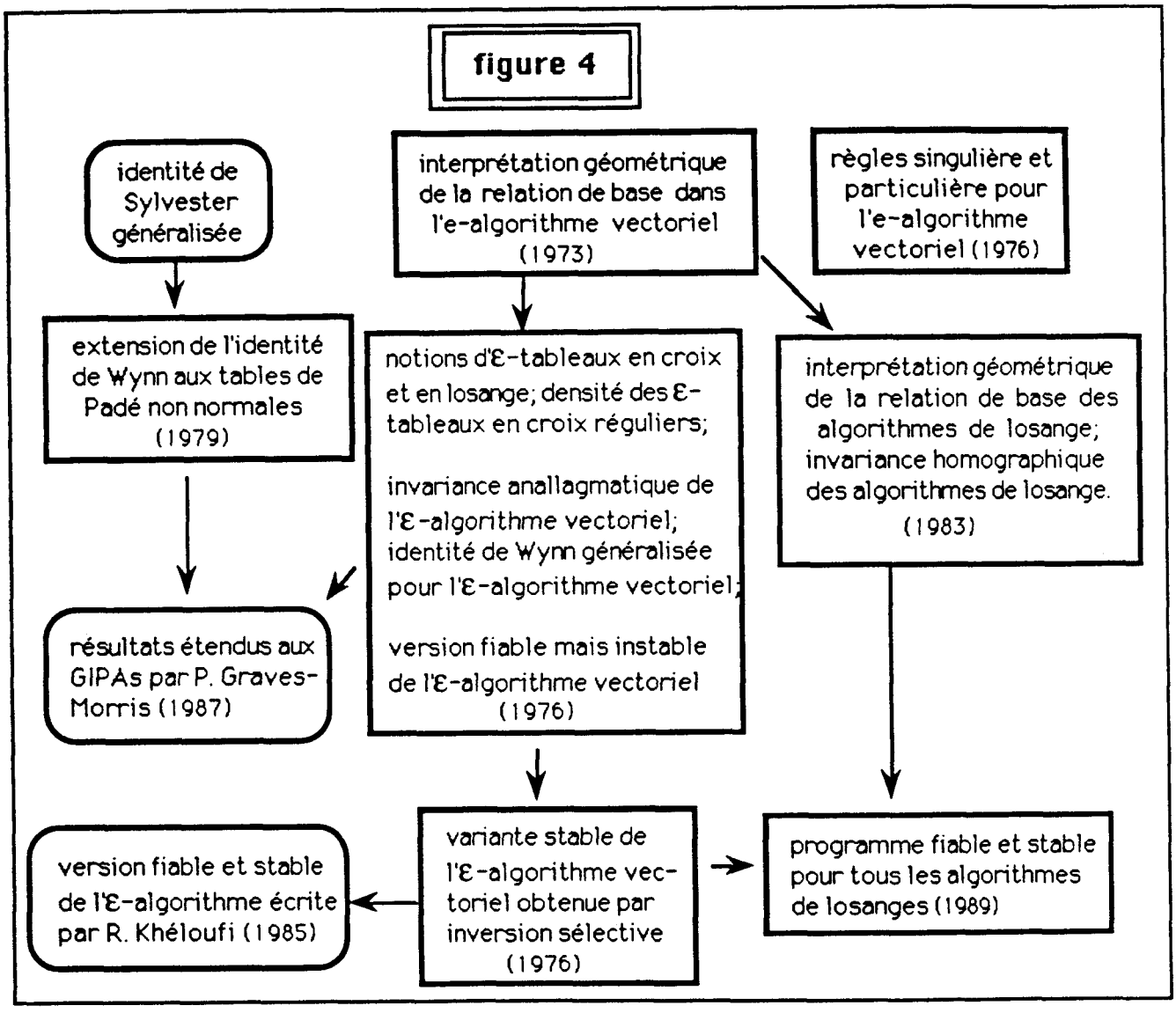
1) Etablissement de règles singulières généralisées pour l'ε-algorithme vectoriel, ce qui conduit à deux résultats (FS2): d'une part, une version fiable mais instable de l'ε-algorithme et d'autre part, une extension de l'identité de Wynn pour la table de Padé scalaire. Cette identité de Wynn généralisée a été récemment étendue à des objets qui généralisent la notion d'approximant de Padé: les approximants de Padé à plusieurs variables par A. Cuyt [35] et les GIPAs ou approximants de Padé vectoriels (ou GIPAs) par P. Graves-Morris et Jenkins [55].

2) Proposition d'une version fiable et numériquement stable de l'ε-algorithme vectoriel (FS3). Présenté au colloque national de Port-Bail en 1976, cet algorithme a été programmé par R.Khéloufi [70] en 1985.

Le fait qu'on sache que, dans le cas de l'ε-algorithme, les singularités ou les quasi-singularités sont circonscrites dans des blos carrés simplifie cette utilisation sélective. La détection et la localisation précise des blocs quasi-singuliers reste encore une question très délicate. Dans le cas des autres algorithmes de losange, l'utilisation sélective est encore plus délicate, voire dangereuse. On lui préfèrera alors une utilisation systématique plus coûteuse certes, mais beaucoup plus fiable.

4.3 *Articulation des papiers.*

La figure 4 rend compte des liaisons entre les documents auxquels je me réfère.



On trouvera dans (FS1) l'interprétation géométrique d'une étape de l'ε-algorithme, interprétation qui a été le point de départ de cette étude. (FS2) rassemble l'ensemble des outils qui permettent l'extension de l'identité de Wynn aux ε-tableaux non normaux. Si certains résultats s'étendent aux autres algorithmes de losange (interprétation géométrique au moyen de birapports, notion de tableau en losange ou en croix, densité des tableaux réguliers dans l'ensemble de tous les tableaux), ce n'est pas le cas de l'identité de Wynn en raison de la présence de blocs singuliers non carrés. (FS3) présente succinctement la première version stable de l'ε-algorithme.

(FS6) comporte deux résultats distincts: un nouvel algorithme de losange dérivé de l'algorithme de Stoer [120] et Larkin [81] mais qu'on peut interpréter comme une extension de l'algorithme de

Thiele [122] et la caractérisation des algorithmes de losange dont les colonnes de même parité sont invariantes dans les transformations homographiques. J'y ai adjoint un exemple illustrant l'intérêt de l'invariance homographique en montrant que cette propriété permettait de résoudre des problèmes d'interpolation rationnelle en présence de blocs singuliers (non carrés). Enfin, (FS7) décrit un programme mettant en œuvre certains algorithmes de losange en exploitant systématiquement l'invariance anallagmatique du tableau construit et une méthode d'estimation des erreurs numériques reposant sur la méthode de perturbation de Vignes.

(FS4) et (FS5) sont totalement dissociés des cinq autres papiers de cette section car ils n'utilisent pas la propriété d'invariance qui est au cœur de ce chapitre: (FS4) est une preuve directe de l'extension de l'identé de Wynn dans le cas de l'ε-algorithme scalaire reposant sur l'identé de Sylvester; les informations sur la table de Padé dont on dispose permettent d'étendre l'identité de Wynn aux tables de Padé non normales. (FS5) présente une règle singulière et une règle particulière permettant de traiter les singularités ou les quasi-singularités isolées dans le cas de l'ε-algorithme vectoriel.

5. Aspects numériques de la programmation.

Les papiers de cette section se rapportent à la programmation des algorithmes numériques et aux questions qui y sont liées: représentation des nombres réels (AN4), mesure de la précision d'un algorithme numérique (AN1), estimation de la précision d'un résultat numérique (AN3) ou validité d'un logiciel numérique (AN2).

On montre en (AN1) comment, à tout calcul fini, on peut associer un nombre qui mesure la stabilité numérique du schéma de calcul mis en œuvre . Tout résultat numérique r d'un tel calcul est entaché d'une erreur δr satisfaisant $|\frac{\delta r}{r}| \leq M_r * \tau$, où τ est une constante attachée au mode de calcul. Sous mon impulsion, cette étude a été reprise dans un contexte probabiliste plus réaliste par H. Nkounkou [102].

Le second papier (AN2) pose la question de la validité du logiciel numérique. Dans une première partie, on tente de sérier les problèmes réels que pose l'introduction de la notion de preuve de programme dans le logiciel numérique. Pour illustrer ce discours, on propose une procédure récursive qui réalise la mise en œuvre adaptative d'une formule de quadrature à laquelle on sait associer une majoration de l'erreur. La présentation d'une telle procédure n'est certes pas nouvelle [111]; ce qui est nouveau, c'est que son utilisation est justifiée dans divers contextes (fonctions monotones, convexes ou concaves), au moyen d'arguments qui prennent en compte plus ou moins complètement les diverses sources d'erreurs numériques.

Le troisième papier (AN3) présente une variante de la méthode du résidu normé de Vignes et La Porte [78] pour apprécier la qualité de la solution d'un système linéaire. L'intérêt de ce papier réside dans le fait qu'on tente de déceler l'ordre de grandeur de l'erreur qui entache la solution à partir des diverses composantes du résidu. L'expérimentation montre que cette méthode permet d'apprécier correctement la précision des composantes les moins entachées d'erreur, mais ne permet guère de déceler la présence de composantes dont la valeur plus petite est entachée d'une erreur relative plus importante.

Le dernier papier (AN4) présente un nouveau mode de représentation des nombres réels dépendant de deux bases a et b en utilisant deux entiers. Une telle représentation repose sur le fait que, pourvu que les deux bases vérifient certaines conditions, l'ensemble $E_{a;b}=\{x=a^p b^q \mid p,q \in \mathbb{Z}\}$ est dense dans $\mathbb{R}$ Ce type de représentation peut s'étendre à d'autres corps (complexes ou quaternions).

6. Liste des documents.

1. Le problème de Fermat-Weber généralisé. (en collaboration avec J.C. Fiorot).

FW1) Trois algorithmes pour résoudre le problème de Fermat-Weber généralisé. Bull. de la direction des études et recherches (E.D.F.). Série C. Suppl.n°2 (1976) 35-54.

FW2) On the Fermat-Weber problem with convex price functions. Surv. of Math. Prog: Proc. of the 9th Int. Math. Prog. Symp. Budapest (1976) 377-408.

FW3) On the Fermat-Weber problem with convex cost functions. Mathematical Programming 14 (1978) 295-311.

2. Etude des Transformations de Suites élémentaires.

TS1) Caractérisation des suites que le procédé θ^2 transforme en suites constantes. C.R.A.S. Paris 284 Série A (1977) 389-392.

TS2) Sur la régularité des procédés δ^2 d'Aitken et W de Lubkin. Padé Approximation and its applications, L. Wuytack ed.,Lecture Notes in Math. 765, Springer Verlag, Berlin Heidelberg New-York (1979) 20-35.

TS3) Une classe de transformations non linéaires de suites de nombres complexes reposant sur l' approximation au sens des moindres carrés. Sémin. d'Anal. Numer. Lille (1977) (Annexe 1).

3. Interpolation Rationnelle.

IR1) Sur une généralisation de l'interpolation rationnelle. Padé Approximation and its applications, Amsterdam 1980, M.G. de Bruin and H. van Rossom ed., Lecture Notes in Maths 888, Springer Verlag, Berlin Heidelberg New-York (1981) 124-135.

IR2) On the use of Kronecker's algorithm in the generalized rational interpolation problem. (Annexe 2).

4. Fiabilité et Stabilité des Algorithmes de Losange.

FS1) Interprétation géométrique d'une étape de l' ε-algorithme. (Annexe 3).

FS2) Règles singulières pour l' ε-algorithme vectoriel. (Annexe 4).

FS3) Une mise en œuvre numériquement stable de l'ε-algorithme vectoriel. Coll. Nat. d'Anal. Num., Port-Bail (1976). (Annexe 5).

FS4) Particular rules for the vector ε-algorithm. Numer. Math. 27 (1977) 203-207.

FS5) Démonstration algébrique de l' extension de l' identité de Wynn aux tables de Padé non normales. Padé Approximation and its applications, L. Wuytack ed., Lecture Notes in Math. 765, Springer Verlag, Berlin Heidelberg New-York (1979) 36-60.

FS6) Utilisation de l' invariance homographique dans les algorithmes de losange. Padé Approximation and its Applications Bad Honnef 1983,H. Werner and H.J. Bünger ed.,Lecture Notes in Math. 1071 (1983) 62-94.

FS7) Une mise en œuvre fiable et stable des algorithmes de losange. (Annexe 6).

5. Aspects Numériques de la Programmation.

AN1) Mesure de la stabilité numérique d'un schéma de calcul sans test. Les Mathématiques de l' Informatique. Colloque de l' A.F.C.E.T. Paris (1982) 505-513.

AN2) Que sait-on prouver en logiciel numérique? Outils, Méthodes et Langages Adaptés au Calcul Scientifique, I.N.R.I.A.,Paris. (1983).

AN3) On the use of the normed residue to check the quality of the solution of a linear system. Proc. of the 11th I.M.A.C.S. Congress. Oslo. Tome1 (1985) 187-191.(1st edition)

AN4) Sur un nouveau mode de représentation des nombres réels. Informatique et Calcul, John Wiley and Sons-Masson, Paris (1986) 44-48.

7. Bibliographie

[1] **Abadie J.**- On the Kuhn-Tucker theorem. Méthods of nonlinear programming, J.Abadie ed., North-Holland, Amsterdam (1967) 19-36.

[2] **Aitken A.C.**- On Bernoulli's numerical solution of algebraic equations. Proc. Roy. Soc. Edinb. 46 (1926) 289-305.

[3] **Alt R.**- Etude numérique de l'erreur numérique d'affectation sur un ordinateur en base quelconque. Rapport I.P. n° 76-5 (1976).

[4] **Alt R.**- Un logiciel pour l'intégration optimale des systèmes. Thèse d'état, Paris (1981).

[5] **Baker G.A. Jr.**- Recursive calculation of Padé approximants. Padé approximants and their applications, P.R. Graves-Morris ed., Academic press, London (1973) 83-91.

[6] **Bauer F.L.**- Computational graphs and rounding errors. S.I.A.M. Journ. Numer. Anal. 11 (1974) 87-96.

[7] **Bossavit A.**- Problèmes du logiciel numérique. Rapport EDF-DER n° HI/3595 02 (1980).

[8] **Brezinski C.**- A general extrapolation algorithm. Numer. Math. 35 (1980) 175-187.

[9] **Brezinski C.**- Accélération de la convergence en analyse numérique. Lecture Notes in Math. 584, Springer Verlag, Berlin Heidelberg New-York (1977).

[10] **Brezinski C.**- Algorithmique numérique. Collection ellipses (1988).

[11] **Brezinski C.**- Conditions d'applications d'application et de convergence de procédés d'extrapolation. Numer. Math. 20 (1972) 64-79.

[12] **Brezinski C.**- Computation of Padé approximants and continued fractions. Journ. of Comput. Appl. Math. 2 (1976) 113-123.

[13] **Brezinski C.**- Computation of the eigenelements of a matrix by the ε-algorithm. Linear Algebra 11 (1975) 7-20.

[14] **Brezinski C.**- Etudes sur les ε-et ρ-algorithmes. Numer. Math. 17 (1971) 153-162.

[15] **Brezinski C.**- Méthodes d'accélération de la convergence en analyse numérique. Thèse d'Etat, Grenoble (1971).

[16] **Brezinski C.**- Rational approximation to formal power series. Journ. Approx. Theory 25 (1979) 295-317.

[17] **Brezinski C.**- Some results in the theory of the vector ε-algorithm. Linear Algebra 8 (1974) 77-86.

[18] **Brezinski C.**- Sur un algorithme de résolution des systèmes non liné-aires. C.R.A.S. Paris 272A (1971) 145-148.

[19] **Bultheel A.**- Division algorithms for continued fractions and the Padé table. Appl. Math. Prog. Div., Kath. Univ. Leuven, Rep. TW41 (1978).

[20] **Bultheel A.**- Fast algorithms for the factorisation of Hankel and Toeplitz matrices and the Padé approximation problem. Appl. Math. Prog. Div., Kath. Univ. Leuven, Rep. TW41 (1978).

[21] **Bultheel A.**- Recursive algorithms for non normal Padé table. Appl. Math. Prog. Div., Kath. Univ. Leuven, Rep. TW40 (1978).

[22] **Bultheel A.**- Recursive algorithms for the Padé table: two approaches. These Procedings.

[23] **Bultheel A.**- Remark on "A new look at the Padé table and different methods for computing its elements". Journ. of Comput. Appl. Math. 5 (1979) 67.

[24] **Bultheel A.**- Recursive algorithms for non normal Padé table. Appl. Math. Prog. Div., Kath. Univ. Leuven, Rep. TW40 (1978).

[25] **Bussonnais B.**- Tous les algorithmes de calcul par récurrence des approximants de Padé d'une série. Construction des fractions continues correspondantes. Exposé au Colloque sur les approximants de Padé, Lille (1978).

[26] **Claessens G., Wuytack L.**- On the computation of non normal Padé approximants. Dep. Wiskunde, Univ. Inst. Antwerpen, Rep. 77-41 (1977).

[27] **Claessens G.**- A new look at the Padé table and different methods for computing its elements. Journ. of Comput. Appl. Math1 (1975) 141-152.

[28] **Claessens G.**- Some aspects of the rational Hermite interpolation table and its applications. Ph. D. thesis, Univ. of Antwerp, 1976.

[29] **Claessens G.**- A useful identity for the rational Hermite interpolation table. Numer. Math. 29 (1978) 227-231.

[30] **Claessens G.**- On the structure of the Newton-Padé table. Journ. of Approx. Theory 22 (1978) 304-319.

[31] **Clark W.D, Gray H.L., and Adams J.E.**- A note on the T-transform of Lubkin. J. Res. N.B.S. 73 B (1969) 25-29.

[32] **Cooper L.**- An extension of the generalized Weber problem. Journ. of Regional Science 8 (1968) 181-197.

[33] **Cooper L.**- Heuristic methods for location-allocation problems. S.I.A.M. Rev. 6 (1964) 37-52.

[34] **Cooper L.**- Location-allocation problems. Operations Research 11 (1963) 331-341.

[35] **Cooper L.**- Solutions of generalized locational equilibrium problem. Journal of Regional Science 7 (1967) 1-18.

[36] **Cuyt A.**- Abstract Padé approximants in operator theory. Padé approximation and its applications, L. Wuytack ed.,Lecture Notes in Math.765, Springer Verlag, Berlin Heidelberg New-York (1979) 61-87.

[37] **Cuyt A.**- Singular rules for the calculation of non-normal multivariate Padé approximants. J. Comp.Appl.Math. 14 (1986) 289-301.

[38] **Dantzig G.B., Wolfe P.**- Decomposition principle for linear programs. Operations Research 8 (1960) 101-111.

[39] **Davis P.J.**- Interpolation and approximation. Dover Pub. Inc., New-York (1975).

[40] **De Boor C.**- Cadre: an algorithm for numerical quadrature. Mathematical software, J.R. Rice ed., Academic Press, New-York and London (1971) 417-449.

[41] **de Montessus de Ballore R.**- Sur les fractions continues algébriques. Bull. Soc. Math. 22 (1902) 28-36.

[42] **Della-Dora.**- Contribution à l'approximation de fonctions de la variable complexe au sens de Hermite, Padé et Hardy. Thèse, Grenoble (1980).

[43] **Dijkstra E.W.**- A Discipline of programming. Prentice Hall (1976).

[44] **Dubovickii A.J., Miljutin A.A.**- Extremum problems in the presence of restrictions. Zh. Vychisl. Mat. mat. Fiz. 5 (1965) 395-453; USSR Comp. Math. and Math. Physics 5 (1965) 1-80.

[45] **Dumontet J.**- Etude de la preuve des programmes numériques: contrôle, prévision et minimisation des erreurs de calcul. Thèse d'Etat, Paris (1979).

[46] **Elion S., Watson-Gandy C.D.T., Christofides N.**- Distribution management: Mathematical modelling and practical analysis. Charles Griffin and C° Ltd, London (1971).

[47] **Espinoza C.**- Mémoire de D.E.A., Univ. de Lille I (1974).

[48] **Floyd R.W.**- Assigning meanings to programs. Proc. Sym. in Applied Math., vol 19, Mathematical aspects of computer science, J.T. Schwartz ed., Amer. Math. Soc.(1967) 19-32.

[49] **Gantmacher F.R.**- The theory of matrices. Vol1. Chelsea, New-York (1960).

[50] **Germain-Bonne B.**- Estimation de la limite de suites et formalisation de procédés d'accélération de la convergence. Thèse (Lille) 1978.

[51] **Germain-Bonne B.**- Transformations de suites. R.A.I.R.O. (1973) 84-90.

[52] **Gilewicz J.**- Approximants de Padé. Lecture Notes in Math. 667, Springer Verlag, Berlin Heidelberg New-York (1978).

[53] **Gragg W.B.**- The Padé table and its relation to certain algorithms of numerical analysis. S.I.A.M. Review 14 (1962) 1-62.

[54] **Graves-Morris P.R.** and **Jenkins C.D.**- Vector-valued rational interpolants III. Constr. Approx. 2 (1986) 263-289.

[55] **Graves-Morris P.R.** and **Jenkins C.D.**- Degeneracies of generalised inverse, vector-valued approximants. A paraître dans Constr. Approx.

[56] **Graves-Morris P.R.**- Efficient reliable rational interpolation. Padé approximation and its applications, Amsterdam 1980, M.G. de Bruin and H. van Rossum ed., Lecture notes in Math. 888, Springer Verlag, Berlin Heidelberg New-York (1981) 28-63

[57] **Graves-Morris P.R.**- The numerical calculation of Padé approximants. Padé approximants and its applications, L. Wuytack ed., Lecture Notes in Math. 765, Springer Verlag, Berlin Heidelberg New-York (1979) 231-245.

[58] **Gray H.L., Clark W.D.**- On a class of nonlinear transformations and their applications to the evaluation of infinite series. Journ. of Res. of the N.B.S. 73B (1969) 251-274.

[59] **Guitel G.**- Histoire comparée des numérations écrites. Editions Flammarion, Paris (1975).

[60] **Hadamard J.**- Récents progrès de la géométrie anallagmatique. Œuvres Tome 2 , Editions du CNRS (1968) 937-986.

[61] **Hardy G.H.** - Divergent series. Clarendon Press, Oxford (1949).

[62] **Hoare C.A.R.**- An axiomatic basis for computer programming. Comm. of the A.C.M. 12 (1969) 576-580,583.

[63] **Huard P.**- Extensions of Zangwill's theorem. Math. Prog. Study 10 (1979) 98-113.

[64] **Ifrah G.**- Histoire universelle des chiffres. Editions Seghers, Paris (1981).

[65] **Jacobsen S.K.**- Weber rides again. Working paper n° 2252/0 37 042, The Institute of Mathematical Statistics and Operations Research, The Technical University of Denmark, Lingby, Denmark (1974).

[66] **Kantorovitch L.V.**- Sur un symbolisme mathématique pour décrire les calculs sur machine. Dokl. Akad. Nauk SSSR 113 (1957) 738-741 (en russe).

[67] **Karlin S.**- Total positivity, vol.1. Stanford Univ. Press, Stanford, Cal. (1968).

[68] **Katz I.N.**- Local convergence in Fermat's problem. Mathematical Programming 6 (1974) 89-104.

[69] **Katz I.N.**- On the convergence of a numerical scheme for solving some locational equilibrium problems. S.I.A.M. J. Appl. Math. 17 (1969) 1224-1231.

[70] **Khéloufi R.**- Sur l'extension de la fiabilité des algorithmes de losange. Thèse de 3ème cycle, Lille (1985).

[71] **Knuth D.E.**- The art of computer programming. Addison-Wesley ed., vol.I: fundamental algorithms, 2nd ed. (1973) and vol.II: seminumerical algorithms, 2nd ed. (1981).

[72] **Kronecker L..**- Zur Theorie des Elimination einer Variabeln aus zwei algebraischen Gleichungen. Monatsber. Koenige. Preuss. Akad. Wiss. Berlin (1881) 535-600.

[73] **Kuenne R.E., Soland R.M.**- Exact and approximate solutions to the multisource Weber problem. Mathematical Programming 3 (1972) 193-209.

[74] **Kuhn H.W., Kuenne R.E.**- An efficient algorithm for the numerical solution of the generalized Weber problem in spacial economics. Journal of Regional Science 4 (1962) 21-33.

[75] **Kuhn H.W.**- A note on Fermat's problem. Mathematical Programming 4 (1973) 98-107.

[76] **Kuhn H.W.**- On a pair of dual nonlinear programs. Methods of nonlinear programming, J. Abadie ed., North Holland, Amsterdam (1967) 38-54.

[77] **Kulisch U.**- An axiomatic approach to rounded computations. Numer. Math. 18 (1971) 1-17.

[78] **La Porte M., Vignes J.**- Algorithmes numériques, analyse et mise en œuvre. Tome 1, Technip, Paris (1974).

[79] **La Porte M., Vignes J.**- Etude statistique des erreurs dans l'arithmétique des ordinateurs; application au contrôle des résultats d'algorithmes numériques. Numer. Math. 23 (1974) 63-72.

[80] **La Porte M. et Vignes J.**- Error analysis computing. Proceedings of IFIP Congress, Stockholm (1974) 610-614.

[81] **Larkin F.M.**- Some techniques for rational interpolation. Computer Journ. 10 (1967) 178-187.

[82] **Larson J.L. Sameh A.H.**- Algorithms for roundoff error analysis. Arelative approach. Computing 24 (1980) 275-297.

[83] **Laurent P.J.**- Approximation et optimisation. Hermann, Paris (1972).

[84] **Levin D.**- Development of non-linear transformation for improving convergence of sequences. Intern. J. Computer Math. B3 (1973) 371-388.

[85] **Lubkin S.**- A method for summing infinite series. J. Res. N.B.S. 48 (1952) 228-254.

[86] **Lyness J.N.**- Notes on the adaptative Simpson Quadrature Routine. Journ. of the A.C.M. 16 (1969) 483-495.

[87] **Maille M.**- Méthodes d'évaluation de la précision d'une mesure ou d'un calcul numérique. Coll. AFCET Les Mathématiques de l'informatique. (1982) 495-503.

[88] **Malcolm M.A., Simpson R.B.**- Local global strategies for adaptative quadrature. A.C.M. Trans. Math. Software 1 (1975) 129-146.

[89] **Marx I.**- Remark concerning a nonlinear sequence-to-sequence transform. J. Math. Phys. 42 (1963) 334-335.

[90] **Mc Leod J.B.**- A note on the ε-algorithm. Computing 7 (1971) 17-24.

[91] **McEliece R.J., Shearer J.B.**- A property of Euclid's algorithm and its application to Padé approximation. S.I.A.M. Journ. Appl. Math. 34 (1978) 611-615.

[92] **McKeeman W.M.**- Algorithm 145, adaptative numerical integration by Simpson's rule. Comm. of the A.C.M. 5 (1962) 604.

[93] **Meinguet J.**- On the solubility of the Cauchy interpolation problem. Approximation theory, A. Talbot ed., Academic Press, London (1970) 137-163.

[94] **Meinguet J.**- Sens et portée des arithmétiques de signification. Colloque Intern. du C.N.R.S., Besançon (1966).

[95] **Miehle W.**- Link-length minisation in networks. Operations Research 6 (1958) 232-243.

[96] **Miller W.**- Software for roundoff analysis. A.C.M. Trans. Math. Soft. 1 (1975) 108-128.

[97] **Minty G.J.**- On the monotonicity of the gradient of a convex function. Pacific Journal of Math. 14 (1964) 243-247.

[98] **Moreau J.J.**- Fonctionnelles sous-différentiables. C.R. Acad. Sci. Paris 257 (1963) 4117-4119.

[99] **Moreau J.J.**- Propriétés des approximations "prox". C.R. Acad. Sci. Paris 256 (1963) 1069-1071.

[100] **Moreau J.J.**- Fonctionnelles convexes. Séminaire sur les équations aux dérivées partielles, Collège de France (1966-1967).

[101] **Muhlbach G.**- The general Neville-Aitken algorithm and some applications. Numer. Math. 31 (1978) 97-110.

[102] **Nkounkou H.**- Contribution à la mesure de la stabilité numérique en moyenne. Thèse de $3^{ème}$ cycle, Univ. de Lille I (1983).

[103] **Overholt K.J.**- Extended Aitken acceleration. B.I.T. 5 (1965) 122-132.

[104] **Pennachi R.**- Le transformazioni razionali di una successione. Calcolo 5 (1968) 37-50.

[105] **Peterson E.**- Symmetric duality for generalized unconstrained geometric programming. S.I.A.M. J. Appl. Math. 19 (1970) 487-526.

[106] **Pichat M.**- Contribution à l'étude des erreurs d'arrondi en arithmétique à virgule flottante. Thèse d'Etat, Grenoble (1976).

[107] **Pichat M.**- Correction d'une somme en arithmétique à virgule flottante. Numer. Math. 19 (1972) 400-406.

[108] **Planchart R., Hurter A.P.**- An efficient algorithm for the solution of the Weber problem with mixed norms. S.I.A.M. J. Control 13 (1975) 650-665.

[109] **Polack P.**- On the implementation of conceptual algorithms. Nonlinear Programming, Mangasarian, Ritter and Rosen ed., Academic Press, New-York (1970) 275-291.

[110] **Rice J.R.**- A theory of condition. S.I.A.M. Journ. Numer. Anal. 3 (1966) 287-310.

[111] **Rice J.R.**- A metalgorithm for adaptative quadrature. Journ. of the A.C.M. 22 (1975) 61-62.

[112] **Rice J.R.**- Sequence transformations based on Tchebycheff approximations. J. Res. N.B.S. 64B (1960) 227-235.

[113] **Rockafellar R.T.**- Characterization of the subdifferentials of convex functions. Pacific Journal of Math. 17 (1966) 497-510.

[114] **Rockafellar R.T.**- Convex analysis. Princeton Univ. Press, Princeton, N.J. (1970).

[115] **Schechter S.**- Minimisation of a convex function by relaxation. Integer and non linear programming, J. Abadie ed., North Holland, Amsterdam (1970) 117-189.

[116] **Shafer R.E.**- On quadratic approximation. S.I.A.M. Journ. Num. Anal. 11 (1974) 447-460.

[117] **Shanks D.**- Nonlinear transformations of divergent and slowly convergent sequences. J. Math.Phys. 34 (1955) 1-42.

[118] **Smith D.A., Ford W.F.**- Acceleration of linear and logarithmic convergence. S.I.A.M. J. Numer.Anal. 16 (1979) 223-240.

[119] **Sterbenz P.H.**- Floating-point computation. Prentice Hall, Englewood Cliffs, N.J. (1974).

[120] **Stoer J.**- Uber zwei Algorithmen zur Interpolation mit Rationalen Funktionen. Numer. Math. 3 (1963) 285-305.

[121] **Streit.**- The T_{+m} transformation. Math. Comp. 30 (1976) 505-511.

[122] **Thiele T.N.**- Interpolations Rechnung. Leipzig (1909).

[123] **Tucker R.R.**- The δ^2-process and related topics. Part I in Pacif. J. Math. 22 (1967) 349-359, Part II in Pacif. J. Math. 28 (1969) 455-463.

[124] **Varaiya P.P.**- Nonlinear programming in Banach spaces. S.I.A.M. J. Appl. Math. 15 (1967) 284-293.

[125] **Vignes J.**- Approche stochastique de l'analyse de propagation des erreurs d'arrondi et de données dans les algorithmes numériques. Ann. des Télécom. 41 (1986) 226-234.

[126] **Warner D.D.**- Hermite interpolation with rational fonctions. Ph. D. Thesis, Univ. of California (1974).

[127] **Weber A.**- Uber den Standort den Industrien. Tübingen (1909).[Translated as "Alfred Weber's theory of the location of industries" by C.J. Friedrich, Univ. of Chicago Press (1929)].

[128] **Weiszfeld E.**- Sur le point pour lequel la somme des distances de n points donnés est minimum. Tôhoku Mathematics Journal 43 (1937) 355-386.

[129] **Wendell R.E., Peterson E.**- Duality in generalized location problems. Working paper, Dept. of Industrial Engineering and Management Sciences, Northwester Univ., Evanston, Ill (1971).

[130] **Werner H.**- A reliable method for rational interpolation. Padé approximation and its applications, L. Wuytack ed., Springer Verlag, Berlin (1979) 257-271.

[131] **Wesolowsky G.O., Love R.F.**- A nonlinear approximation method for solving a generalized rectangular distances Weber problem. Management Science 18 (1972) 656-664.

[132] **Wesolowsky G.O., Love R.F.**- The optimal location of new facilities using rectangular distances. Operations research 19 (1971) 124-130.

[133] **Wilkinson J.H.**- Rounding error in algebraic processes. Her Majesty's stationery office, London (1963).

[134] **Wimp J.**- Some transformations of monotone sequences. Math. Comp. 26 (1972) 251-254.

[135] **Wirth N.**- Program development by stepwise refinement. Comm. of the A.C.M. 14 (1971) 221-227.

[136] **Wuytack L.**- On the osculatory rational interpolation problem. Math. Comp. 29 (1975) 837-843.

[137] **Wynn P.**- A numerical method for estimating parameters in mathematical models. Centre de Recherches Mathématiques, Université de Montréal, CRM-443 (1974).

[138] **Wynn P.**- A sufficient condition for the instability of the ε-algorithm. Nieuw. Arch. Wisk. 3 (1961) 117-119.

[139] **Wynn P.**- Acceleration techniques for iterated vector and matrix problems. Math. Comp. 16 (1962) 301-322.

[140] **Wynn P.**- L'ε-algorithma e la tavola di Padé. Rend. di Mat. Roma 20 (1961) 403-408.

[141] **Wynn P.**- On a device for computing the $e_m(S_n)$ transformation. M.T.A.C. 10 (1956) 91-96.

[142] **Wynn P.**- On a procrustean technique for the numerical transformation of slowly convergent sequences and series. Proc. Camb. Phil. Soc. 52 (1956) 663-671.

[143] **Wynn P.**- Singular rules for certain nonlinear algorithms. B.I.T. 3 (1963) 175-195.

[144] **Wynn P.**- Upon systems of recursions which obtain among the quotients of the Padé table. Numer. Math. 8 (1966) 264-269.

[145] **Zangwill W.I.**- Nonlinear programming: an unified approach. Prentice Hall, Englewood Cliffs, N.J. (1969).

[146] **Zoutendijk G.**- Methods of feasible directions. Elsevier, Amsterdam (1960). Proc. Roy. Soc. Edinb. 46 (1926) 289-305.

Une classe de transformations non linéaires de suites de nombres complexes reposant sur l'approximation au sens des moindres carrés.

Résumé

On propose une nouvelle classe de transformations non linéaires de suites de nombres complexes reposant sur l'approximation au sens des moindres carrés. Ces transformations notées $C_k^{(m)}$ (avec $m \geq 2k$) généralisent les classiques transformations E_k de Shanks (dont l'ε-algorithme assure une mise en œuvre efficace) puisqu'on a: $C_k^{(2k)} = E_k$. Trois aspects fondamentaux de la plus simple de ces transformations, le procédé $C_1^{(m)}$ qui généralise le procédé δ^2 d'Aitken, sont alors étudiés: on montre que ce procédé conserve la limite d'une suite dès qu'il conserve sa convergence et que son action sur les suites à convergence géométrique a pour effet d'accélérer la convergence d'une part, et de ne pas amplifier exagérément les éventuelles perturbations de la suite initiale d'autre part.

Annexe 1: transformation de suites par approximation.

1. Introduction.

La plupart des procédés d'accélération de la convergence de suites actuellement connus reposent sur des techniques d'interpolation des valeurs de la suite par une fonction d'un type choisi [2]. Le principal inconvénient de ces procédés est leur sensibilité à la précision de la suite initiale: ils sont généralement instables en ce sens qu'une légère perturbation de la suite donnée modifie notablement les estimations de la limite que fournissent les procédés en question. Cela provient du fait que l'interpolation est par essence une technique numériquement instable et les procédés reposant sur cette technique ne peuvent pas échapper à cet inconvénient.

Pour remédier à cette déficience, on peut alors rechercher des procédés qui reposent sur l'approximation, technique dont la stabilité numérique est nettement meilleure que celle de l'interpolation. Malheureusement, les outils dont nous disposons à cette fin sont nettement moins commodes que ceux que nous procure l'interpolation et c'est probablement pourquoi de telles études n'ont été que rarement entreprises [9].

Le second paragraphe est consacré à la présentation d'un procédé répondant à ces exigences. Après avoir envisagé un certain nombre de façons de généraliser la transformation de Shanks [11] en des termes d'approximation, nous en retenons une qui repose sur l'approximation au sens des moindres carrés. Nous obtenons ainsi une classe de transformations dépendant de deux indices k et m (avec $2 \leq 2k \leq m$), transformations qu'on notera $C_k^{(m)}$ et qui vérifient $C_k^{(2k)} = E_k$, où E_k est la classique transformation de Shanks, et nous montrerons que ces procédés sont translatifs et homogènes [5].

La suite du papier est consacrée à une étude de la plus simple de ces transformations, le procédé $C_1^{(m)}$ qui est une généralisation du procédé δ^2 d'Aitken. Trois aspects essentiels de cette transformation sont successivement étudiés: la conservation de la limite, l'accélération de la convergence, et la sensibilité aux perturbations de la suite initiale. Dans le troisième paragraphe, il est montré que, si le procédé préserve la convergence d'une suite, alors il conserve la limite. Cette propriété, qui généralise un résultat de Lubkin relatif au procédé δ^2, est appelée "join convergence" par Rice [9] et n'a été établie que pour un nombre très limité de transformations non linéaires: la transformation C de Rice [9], le procédé T_{11} de Gray et Clarke [6] par Streit [12] et le procédé W de Lubkin [7], première étape du θ-algorithme de Brezinski [1] par l'auteur [3].

Pour les deux autres questions, nous avons limité l'étude du procédé au cas où il agit sur des suites à convergence géométrique. Dans le quatrième paragraphe, nous montrons que, tel le procédé δ^2 qu'il généralise, ce procédé $C_1^{(m)}$ accélère la convergence de toute suite à convergence géométrique. Le cinquième paragraphe est consacré à l'étude du conditionnement du procédé vis à vis des suites à convergence géométrique: après avoir défini le nombre de conditionnement d'une transformation de suites par rapport à une suite, nous montrons que le nombre de conditionnement des procédés $C_1^{(m)}$ par rapport aux suites à convergence géométrique est arbitrairement voisin de l'unité pourvu que l'on retienne une valeur du paramètre m assez grande. Quelques exemples numériques viennent appuyer ce dernier point et nous permettre de conclure.

2. Les transformations $C_k^{(m)}$

2.1 *Rappel de la transformation de Shanks [11]*

La transformation de Shanks $E_k(x_n)$ de la suite $(x_n)_{n \geq 0}$ peut être définie de la façon suivante:

Annexe 1: transformation de suites par approximation.

Etant donnés 2k+1 termes consécutifs x_{n+i} *(i=0,...,2k) de la suite initiale, on recherche s'il existe une suite* $(y_n)_{n\geq 0}$ *vérifiant les deux conditions:*

(L) *Condition de* ***récurrence linéaire****:*
(y_n) *satisfait une équation de récurrence linéaire d'ordre k à coefficients constants de somme non nulle:*

$$\sum_{i=0}^{k} \rho_i (y_{n+i} - \hat{y}), \ \forall n \geq 0, \quad \text{avec} \sum_{j=0}^{k} \rho_j \neq 0.$$

(I) *Condition d'****interpolation****:*
La restriction de (y_n) *aux indices n+i (i=0,...,2k) s'identifie à celle de* (x_n)*:*

$$y_{n+i} = x_{n+i}, \quad i=0,...,2k.$$

Notons Δ l'opérateur qui, à la suite u de termes u_n associe la suite v de terme général $v_n = u_{n+1} - u_n$, et posons: $\Delta u_n = v_n$. On a donc: $\Delta^2 u_n = u_{n+2} - 2u_{n+1} + u_n$. Si nous notons $H_k(u_n)$ le déterminant de la matrice U d'ordre k dont le terme général U_i^j est égal à $u_{n+i+j-2}$ pour i,j=1,...,k (matrice de Hankel), nous savons que le problème précédent admet une solution unique si et seulement si $H_k(\Delta^2 x_n)$ est non nul. *Le transformé* $E_k(x_n)$ *est alors par définition la limite (ou l'anti-limite [11])* $\hat{y}$ *de la suite interpolante* (y_n) *et on vérifie aisément que:* $E_k(x_n) = H_k(x_n)/H_k(\Delta^2 x_n)$.

Il est possible de calculer ces quotients de déterminants sans devoir calculer les déterminants eux-mêmes en utilisant un procédé récurrent introduit par Wynn [14], l'ε-algorithme.

Remarquons que, dans la condition (L), la contrainte $\sum_{j=0}^{k} \rho_j \neq 0$ peut être remplacée par $\sum_{j=0}^{k} \rho_j = 1$ et rappelons que la transformation E_1 n'est autre que le classique procédé δ^2 d'Aitken.

2.2 *Généralisations de la transformation de Shanks par approximation.*

2.2.1 Motivation.

La transformation de Shanks est donc un procédé reposant sur l'interpolation d'une suite donnée par une suite vérifiant une équation de récurrence linéaire à coefficients constants de somme non nulle. L'efficacité de ce procédé a été confirmée par l'expérience [16]. Toutefois cette efficacité pâtit de l'instabilité attachée à toutes les méthodes reposant sur l'interpolation. Pour s'en con-vaincre, le lecteur peut consulter les résultats numériques fournis par l'application du procédé δ^2 à une suite perturbée (voir le chapitre 6). Il apparaît clairement qu'une légère perturbation de la suite initiale peut modifier très sensiblement les estimations de la limite fournies par le procédé δ^2. Cette constatation nous conduit à tenter de définir des transformations qui généralisent la transformation Shanks (pour en conserver l'efficacité théorique) en remplaçant le contexte de l'interpolation par celui de l'approximation (pour améliorer la stabilité).

2.2.2 Démarche générale.

Supposons que nous disposions d'un échantillon de m+1 valeurs consécutives x_i (i=0,1,...,n) d'une suite que nous nous proposons de transformer. Considérons une suite finie de nombres réels $(y_i)_{i=0...m}$, un nombre réel $\hat{y}$, et une autre suite finie de réels $(\rho_j)_{j=0...k}$ de nombres complexes avec $m \geq 2k$, et associons leur les nombres:

$$r_i = \sum_{j=0}^{k} \rho_j (y_{i+j} - \hat{y}) \quad (\text{pour } i=0 \ldots m-k)$$

$$\eta = \sum_{j=0}^{k} \rho_j - 1$$

$$\varepsilon_i = y_i - x_{n+i} \quad (\text{pour } i=0 \ldots m)$$

La transformation E_k de Shanks consiste à imposer $m=2k$ et à rechercher les nombres y_i $(i=0,\ldots,2k)$, $\hat{y}$, et ρ_j $(j=0,\ldots,k)$ de façon que:

(1) $r_i=0$, $i=0,\ldots,k$,

(2) $\eta=0$,

(3) $\varepsilon_i=0$, $i=0,\ldots,k$.

Si nous supposons désormais que $m>2k$, il n'est généralement plus possible de trouver des nombres y_i, $\hat{y}$, et ρ_j vérifiant toutes ces conditions. Nous pouvons alors nous contenter d'imposer la satisfaction d'un sous-ensemble d'entre elles et de minimiser une norme du vecteur dont les composantes sont les résidus associés aux contraintes non satisfaites. Cette optique conduit à un grand nombre de transformations de suites qui se réduisent toutes à la transformation E_k dans le cas particulier où $m=2k$ et seront autant de généralisations de cette transformation. Ces nouvelles transformations dépendent de deux choix:

1) le choix du sous-ensemble des contraintes satisfaites;
2) le choix de la norme des résidus associés aux autres contraintes.

2.2.3 Choix de contraintes et de normes.

2.2.3.1 Le choix des contraintes.

Parmi les choix possibles, il en est deux qui retiennent plus particulièrement l'attention:

1° le choix le plus naturel consiste certainement à conserver les contraintes (1) et (2) et à minimiser une norme ϕ du vecteur ε. Cela revient à rechercher parmi les suites vérifiant une équation de récurrence linéaire d'ordre k à coefficients constants et de somme égale à un, celle qui est la plus voisine de l'échantillon $(x_{n+i})_{i=0\ldots m}$ au sens de la norme ϕ. Des méthodes issues d'un tel choix ont été étudiées par Rice [9].

2° Un choix moins naturel mais conduisant à une analyse plus commode consiste à conserver les contraintes (2) et (3) et à minimiser une norme ϕ du vecteur r. Cela revient à interpoler la suite (x_n) aux m+1 points x_{n+i} par une suite vérifiant au mieux au sens de la norme ϕ une équation de récurrence du type (L). La méthode présentée en 2.3 est issue de ce choix.

2.2.3.2 le choix des normes.

Nous nous limiterons ici aux deux normes classiques suivantes:

1° la norme euclidienne ϕ_2: $x \in \mathbb{R}^n \rightarrow \phi_2(x) = \left(\sum_{i=1}^{p} (x_i)^2\right)^{1/2}$.

2° la norme du maximum ϕ_∞: $x \in \mathbb{R}^n \rightarrow \phi_\infty(x) = \max_{i=1\ldots p} |x_i|$.

2.2.4 les procédés de Rice.

Si nous retenons le premier choix de contraintes et la norme du maximum, nous nous ramenons au problème suivant:

minimiser ε *sous les contraintes suivantes:*

$$|y_i - x_{n+i}| \le \varepsilon, \quad i=0,\dots,m.$$

$$\sum_{j=0}^{k} \rho_j (y_{i+j} - \hat{y}) = 0, \quad \text{pour } i=0,1,\dots,m-k,$$

$$\sum_{j=0}^{k} \rho_j = 1.$$

Ce problème a été abordé par Rice [9] dans le cas où $m=2k+1$ pour les deux premières valeurs de k en se limitant aux suites réelles. Dans le cas où $k=1$, la condition imposée à la suite (y_i) équivaut à: $y_i = \hat{y} + b\lambda^i$ pour $i \ge 0$, où $\hat{y}$, b, λ sont réels, et Rice montre que la solution du problème modifié est fournie par: $\hat{y} = E_1(u_n)$, où $u_n = (x_n + x_{n+1})/2$, et où E_1 représente le procédé δ^2 d'Aitken.
Dans le cas où $k=2$, la condition imposée à la suite (y_i) revêt l'une des deux formes suivantes:

$$y_i = \hat{y} + a\lambda^i \cos(\theta + i\varphi), \quad \text{avec } \hat{y}, a, \theta, \varphi \in \mathbb{R},$$

$$y_i = \hat{y} + (a+ib)\lambda^i, \quad \text{avec } \hat{y}, a, \lambda, b \in \mathbb{R}.$$

En fait, Rice se contente d'envisager la première condition et il conjecture alors que: $\hat{y} = E_2(u_n)$, où $u_n = (x_n + x_{n+1})/2$. Les deux procédés proposés par Rice consistent donc en la composition de deux transformations classiques: une transformation linéaire (Holder, Cesaro, ou Euler) et la transformation de Shanks.

2.3 Les transformations $C_k^{(m)}$

Si nous retenons le second choix de contraintes et la norme euclidienne, nous sommes ramenés au problème suivant:

$$(P_1) \quad \begin{cases} \textit{minimiser} & \displaystyle\sum_{i=0}^{m-k} \Big| \sum_{j=0}^{k} (\rho_j x_{n-j}) - \hat{x} \Big|^2 \\ \textit{sous la contrainte:} & \displaystyle\sum_{j=0}^{k} \rho_j = 1. \end{cases}$$

En vue d'alléger la présentation des résultats, nous introduisons les matrices suivantes:

$$X^{(n)} = \begin{pmatrix} x_n & \dots & x_{n+k} \\ \vdots & & \vdots \\ x_{n+m-k} & \dots & x_{n+m} \end{pmatrix}, \Delta^{(n)} = \begin{pmatrix} \Delta x_n & \dots & \Delta x_{n+k-1} \\ \vdots & & \vdots \\ \Delta x_{n+m-k} & \dots & \Delta x_{n+m-1} \end{pmatrix}, x^{(n)} = \begin{pmatrix} x_n \\ \vdots \\ x_{n+m-k} \end{pmatrix}, \rho = \begin{pmatrix} \rho_0 \\ \vdots \\ \rho_k \end{pmatrix},$$

et nous notons u et v les vecteurs respectifs de $\mathbb{C}^{m-k+1}$ et $\mathbb{C}^{k+1}$ dont toutes les composantes valent 1.
En outre, la matrice transposée et conjuguée d'une matrice M sera notée $\tilde{M}$.

Dans ces conditions, le problème (P_1) s'écrit encore:

$$(P_2) \quad \begin{cases} \min \|X^{(n)}\rho - \hat{x}\| \\ \text{sous la contrainte: } \tilde{v}\rho = 1. \end{cases}$$

et on a le résultat suivant:

Annexe 1: transformation de suites par approximation.

Proposition 1: *Pour que le problème (P_1) admette une solution unique, il faut et il suffit que les m+1 valeurs x_{n+i} (i=0,...,m) soient telles que le système homogène:*

$$\beta_0+\sum_{j=1}^{k}\beta_j\Delta x_{n+i+j-1}=0, \quad (i=0,...,m-k)$$

n'ait pas d'autre solution que la solution triviale: $\beta_j=0$, j=0,...,k.

Démonstration: Grace à la contrainte $\tilde{v}\rho=1$, l'expression $s_{n+i}=\sum_{j=0}^{k}(\rho_j x_{n+i+j})-\hat{x}$ s'écrit encore:

$$s_{n+i}=x_{n+i}-\hat{x}+\sum_{j=0}^{k}\rho_j(x_{n+i+j}-x_{n+i}).$$

Puisque $x_{n+i+j}-x_{n+i}=\sum_{l=1}^{j}\Delta x_{n+i+l-1}$, nous obtenons, après permutation des symboles de sommation:

$$\sum_{j=0}^{k}\rho_j(x_{n+i+j}-x_{n+i})=\sum_{l=1}^{k}(\Delta x_{n+i+l-1}\sum_{j=1}^{k}\rho_j).$$

Si nous posons: $\sigma_l=-\sum_{j=1}^{k}\rho_j$, le problème (P_1) devient le problème sans contrainte suivant :

$$(P_3) \qquad \min \sum_{i=0}^{m-k}|x_{n+i}-\hat{x}-\sum_{l=1}^{k}(\Delta x_{n+i+l-1}\sigma_l)|^2,$$

qui n'est autre que celui de la minimisation de la norme euclidienne du résidu attaché au système linéaire surabondant:

$$\hat{x}+\sum_{l=1}^{k}(\Delta x_{n+i+l-1}\sigma_l)=x_{n+i} \quad (i=0,...,m-k).$$

Si nous notons σ le vecteur de composantes σ_l (i=1,...,k), ce problème (P_3) s'écrit encore:

$$(P_4) \qquad \min \| (u,\Delta X^{(n)})\begin{pmatrix}\hat{x}\\ \Delta X^{(n)}\end{pmatrix}-x^{(n)}\|^2,$$

et on sait que ce problème a une solution unique si et seulement si le système linéaire:

$$(L) \qquad \begin{pmatrix}\tilde{u}\\ \tilde{\Delta X^{(n)}}\end{pmatrix}[(u,\Delta X^{(n)})\begin{pmatrix}\hat{x}\\ \Delta X^{(n)}\end{pmatrix}-x^{(n)}] = 0$$

a une solution unique. Ceci est réalisé si et seulement si la matrice $(u, \Delta X^{(n)})$ est de rang maximal k+1. Pour achever la démonstration, il suffit de noter que la condition nécessaire et suffisante pour que la matrice soit de rang inférieur à k est que ses colonnes soient linéairement dépendantes.

Corollaire: *Si le problème (P_1) admet une solution unique $\hat{x}$, elle vérifie:*

$$(S) \qquad \hat{x}=\frac{det\begin{bmatrix}\tilde{u}\\ \tilde{\Delta X^{(n)}}\end{bmatrix}[(x,\Delta X^{(n)})]}{det\begin{bmatrix}\tilde{u}\\ \tilde{\Delta X^{(n)}}\end{bmatrix}[(u,\Delta X^{(n)})]}$$

Démonstration: Il suffit de noter que, dans ce cas, la solution de (P_1) est identique à celle du système linéaire (L). La première composante $\hat{x}$ a alors la valeur annoncée.

Définition: *A tout $(m+1)$-uple $(x_{n+i} / i=0,...,m)$ tel que le problème (P_1) admette une solution, nous associons $C_k^{(m)}(x_n)=\hat{x}$, où $\hat{x}$ est la solution unique définie par (S). La transformation de suite associée sera notée $C_k^{(m)}$: $(x_n)_{n\geq 0} \rightarrow (y_n)_{n\geq 0}$ définie par: $y_n=C_k^{(m)}(x_n)$.*

2.4 Propriétés élémentaires des transformations $C_k^{(m)}$.

L'étude générale des procédés $C_k^{(m)}$ est assez délicate et nous nous contenterons de présenter ici trois propriétés très élémentaires.

2.4.1 Homogénéité et translativité.

Définition [5] : *Une transformation de suite T sera dite:*

$$\left.\begin{array}{l} \text{homogène} \Leftrightarrow T(ax_n)=aT(x_n) \\ \text{translative} \Leftrightarrow T(x_n+a)=T(x_n)+a \end{array}\right\} \forall a \in \mathbb{C}, \forall n \in N.$$

Proposition 2: *Les transformations $C_k^{(m)}$ sont homogènes et translatives.*

La **démonstration** de ces propriétés est immédiate: elle repose sur la linéarité d'un déterminant par rapport aux termes d'une ligne ou d'une colonne pour la première, et sur le fait que $\Delta(x_n+a)=\Delta x_n$ pour la seconde.

2.4.2 Existence d'un transformé.

lemme. *Si les $m+1$ nombres x_i $(i=0,...,m)$ sont tels qu'il existe un entier $q \leq k$, des coefficients $\gamma_j^{(q-1)}$ $(j=0,...,k-q+1)$, de somme nulle, et un polynôme Q_{q-1} de degré $q-1$ au plus tels que:*

$$\sum_{j=0}^{k-q+1} \gamma_j^{(q-1)} x_{n+i+j} = Q_{q-1}(n+i) \quad (\text{pour } i=0,...,m-k+q-1),$$

alors il existe des coefficients $\gamma_j^{(q)}$ $(j=0,...,k-q)$ et un polynôme Q_q de degré q au plus tels que:

$$\sum_{j=0}^{k-q} \gamma_j^{(q)} x_{n+i+j} = Q_q(n+i), \quad (\text{pour } i=0,...,m-k+q).$$

Preuve: Grace à la nullité de la somme des coefficients, nous avons en effet:

$$\sum_{j=0}^{k-q+1} \gamma_j^{(q-1)} x_{n+i+j} = \sum_{j=0}^{k-q+1} \gamma_j^{(q-1)} \left(x_{n+i} + \sum_{l=1}^{j} \Delta x_{n+i+l-1}\right) = \sum_{j=1}^{k-q+1} \gamma_j^{(q-1)} \sum_{l=1}^{j} \Delta x_{n+i+l-1}.$$

Après permutation des symboles de sommation, il vient:

$\sum_{j=0}^{k-q+1} \gamma_j^{(q-1)} x_{n+i+j} = \sum_{l=1}^{k-q+1} \Delta x_{n+i+l-1} \sum_{j=l}^{k-q+1} \gamma_j^{(q-1)}$, d'où, en posant $\gamma_l^{(q)} = \sum_{j=l+1}^{k-q+2} \gamma_j^{(q-1)}$:

$\sum_{j=0}^{k-q+1} \gamma_j^{(q-1)} x_{n+i+j} = \sum_{j=0}^{k-q} \gamma_j^{(q)} \Delta x_{n+i+j} = \Delta(\sum_{j=0}^{k-q} \gamma_j^{(q)} x_{n+i+j})$, où Δ agit sur le seul indice n.

Tout polynôme Q_{q-1} de degré q-1 est tel qu'il existe un polynôme Q_q de degré q tel que:

$Q_{q-1}(n) = \Delta Q_q(n)$, $\forall n \in \mathbb{N}$. La condition $\sum_{j=0}^{k-q+1} \gamma_j^{(q-1)} x_{n+i+j} = Q_{q-1}(n+i)$ équivaut donc à:

$\Delta(\sum_{j=0}^{k-q} \gamma_j^{(q)} x_{n+i+j} - Q_q(n+i)) = 0$, et il existe donc une constante réelle K telle que l'on ait, pour

$i=0,\ldots,m-k+q$: $\sum_{j=0}^{k-q} \gamma_j^{(q)} x_{n+i+j} - Q_q(n+i) = K$.

On achève la preuve du lemme en intégrant la constante K au polynôme Q_q.

La condition introduite dans la proposition 1 peut alors être affinée par le résultat suivant:

Proposition 3 *Une condition nécessaire et suffisante pour que le transformé $C_k^{(m)}(x_n)$ ne soit pas défini (pour une valeur précise des entiers k,m et n) est qu'il existe un entier positif $p \leq k$, des nombres complexes γ_l (l=0,...,k-p) de somme non nulle, et un polynome P de degré inférieur ou égal à p tel que:*

$$\sum_{l=0}^{k-p} \gamma_l x_{n+i+l} = P(n+i) \quad (\text{pour } i=0,\ldots,m-k+p).$$

Démonstration: La condition nécessaire et suffisante de la proposition 1:

$$\beta_0 + \sum_{j=1}^{k} \beta_j \Delta x_{n+i+j-1} = 0, \quad (i=0,\ldots,m-k)$$

s'écrit encore: $\Delta [\beta_0(n+i) + \sum_{j=1}^{k} \beta_j x_{n+i+j-1}] = 0$, $(i=0,\ldots,m-k)$, soit:

$\sum_{j=0}^{k-1} \beta_{j+1} \Delta x_{n+i+j} = P_1(n+i)$, $(i=0,\ldots,m-k+1)$, où P_1 est un polynôme de degré inférieur ou égal à un.

Le lemme précédent nous apprend que, si la somme des poids de la relation de récurrence est nulle, il existe une relation de récurrence d'ordre inférieur dans laquelle le degré du polynôme du second membre est supérieur d'une unité. L'utilisation de ce lemme sera itérée jusqu'à obtention d'un ensemble de coefficients de somme non nulle. Pour achever la démonstration, nous remarquerons que, si q=k, on a nécessairement $\gamma_0^{(q)} \neq 0$, ce qui stoppe la récurrence. Dans le cas contraire, cela signifierait que la condition initiale se ramène à l'identité 0=0, ce qui implique la nullité de tous les β_j.

Remarque: Nous venons de montrer que le transformé $C_k^{(m)}(x_n)$ n'est pas défini dans le seul cas où la suite finie $(x_{n+i})_{i=0\ldots m}$ vérifie une relation de récurrence avec un second membre polynomial, la

somme de l'ordre k-p de la relation de récurrence et du degré du polynôme étant au plus égal à k. Dans le cas particulier où le degré du polynôme est nul, la relation de récurrence s'écrit:

$$\sum_{j=0}^{k-p}\gamma_{j+1}^{(p)}\Delta x_{n+i+j}=K,\ (i=0,\dots,m-k+p)\ \text{avec}\ \sigma=\sum_{j=0}^{k-p}\gamma_{j+1}^{(p)}\neq 0,$$

et on aura alors: $C_{k-p}^{(m')}(x_{n+i})=K/\sigma$ pour $i=0,\dots,m-m'$, et pour tout m' vérifiant: $2(k-p)\leq m'\leq m$.

Ceci signifie qu'il existe un procédé d'ordre inférieur qui fournit la limite ou l'anti-limite [11] d'une suite interpolant les m+1 valeurs x_{n+i}.

Par contre, si le degré du second membre est effectivement non nul, aucune suite interpolante vérifiant une relation de récurrence d'ordre inférieur ou égal à k n'a de limite ou d'anti-limite finie et le procédé est alors en échec.

2.4.3 Qualité de l'approximation.

Puisque le procédé se contente d'approcher la suite vérifiant une équation de récurrence linéaire, il est naturel de chercher à estimer a posteriori la qualité de cette approximation.

Proposition 4: $$\sum_{i=0}^{m-k}\Big|\sum_{j=0}^{k}(\rho_j x_{n-j})-C_k^{(m)}(x_n)\Big|^2=\det\left(\begin{pmatrix}\tilde{u}\\ \tilde{X}^{(n)}\end{pmatrix}(u,X^{(n)})\right)\Big/\det\left(\begin{pmatrix}\tilde{u}\\ \tilde{\Delta X}^{(n)}\end{pmatrix}(u,\Delta X^{(n)})\right).$$

Démonstration: Un calcul élémentaire montre qu'une condition nécessaire et suffisante pour que le point $(\hat{x},\rho)$ soit stationnaire pour le problème (P_1) est qu'il existe $\lambda\in\mathbb{C}$ tel que:

$$\begin{pmatrix}-\tilde{u}u & \tilde{u}X^{(n)} & 0\\ -\tilde{X}^{(n)}u & \tilde{X}^{(n)}X^{(n)} & v\\ 0 & v & 0\end{pmatrix}\begin{pmatrix}\hat{x}\\ \rho\\ \lambda\end{pmatrix}=\begin{pmatrix}0\\ 0\\ 1\end{pmatrix}$$

Le déterminant D de ce système s'écrit:

$$D=\begin{vmatrix}-\tilde{u}u & \tilde{u}X^{(n)} & 0\\ -\tilde{X}^{(n)}u & \tilde{X}^{(n)}X^{(n)} & v\\ 0 & v & 0\end{vmatrix}=\begin{vmatrix}-\tilde{u}u & \tilde{u}x^{(n)} & \tilde{u}\Delta X^{(n)} & 0\\ -\tilde{X}^{(n)}u & \tilde{X}^{(n)}x^{(n)} & \tilde{X}^{(n)}\Delta X^{(n)} & v\\ 0 & 1 & 0 & 0\end{vmatrix}$$

Des manipulations élémentaires montrent que:

$$D=\begin{vmatrix}-\tilde{u}u & \tilde{u}x^{(n)} & \tilde{u}\Delta X^{(n)} & 0\\ -\tilde{x}^{(n)}u & \tilde{x}^{(n)}x^{(n)} & \tilde{x}^{(n)}\Delta X^{(n)} & 1\\ -\Delta\tilde{X}^{(n)}u & \Delta\tilde{X}^{(n)}x^{(n)} & \Delta\tilde{X}^{(n)}\Delta X^{(n)} & 0\\ 0 & 1 & 0 & 0\end{vmatrix}=\begin{vmatrix}\tilde{u}u & \tilde{u}\Delta X^{(n)}\\ \Delta\tilde{X}^{(n)}u & \Delta\tilde{X}^{(n)}\Delta X^{(n)}\end{vmatrix}$$

D'où: $D=\det\left(\begin{pmatrix}\tilde{u}\\ \tilde{\Delta X}^{(n)}\end{pmatrix}(u,\Delta X^{(n)})\right)$.

D'autre part, calculons le déterminant N défini par:

$$N=\begin{vmatrix} -\tilde{u}u & \tilde{u}X^{(n)} & 0 \\ -\tilde{X}^{(n)}u & \tilde{X}^{(n)}X^{(n)} & 0 \\ 0 & v & 1 \end{vmatrix} = -\begin{vmatrix} \tilde{u}u & \tilde{u}X^{(n)} \\ \tilde{X}^{(n)}u & \tilde{X}^{(n)}X^{(n)} \end{vmatrix} = \det\left\{\begin{pmatrix} \tilde{u} \\ \tilde{X}^{(n)} \end{pmatrix}(u,X^{(n)})\right\}.$$

Le multiplicateur de Lagrange λ associé à la contrainte $\tilde{v}\rho=1$ a donc pour valeur:

$$\lambda=N/D=\det\left\{\begin{pmatrix} \tilde{u} \\ \tilde{X}^{(n)} \end{pmatrix}(u,X^{(n)})\right\}/\det\left\{\begin{pmatrix} \tilde{u} \\ \tilde{\Delta X}^{(n)} \end{pmatrix}(u,\Delta X^{(n)})\right\}.$$

Puisque $\tilde{v}\rho=1$, à l'optimum, nous avons:

$$\|X^{(n)}\rho-\hat{x}u\|^2=\|X^{(n)}\rho-\hat{x}u\tilde{v}\rho\|^2=\|(X^{(n)}-\hat{x}u\tilde{v})\rho\|^2=\tilde{\rho}(\tilde{X}^{(n)}-\hat{x}u\tilde{v})(X^{(n)}-\hat{x}u\tilde{v})\rho$$
$$=\tilde{\rho}\tilde{X}^{(n)}(X^{(n)}-\hat{x}u\tilde{v})\rho-\hat{x}\tilde{\rho}v\tilde{u}(X^{(n)}-\hat{x}u\tilde{v})\rho.$$

Cette identité $\tilde{v}\rho=1$ permet encore d'écrire:

$$\tilde{u}(X^{(n)}\rho-\hat{x}u)=0\Rightarrow\tilde{\rho}v\tilde{u}(X^{(n)}-\hat{x}u\tilde{v})\rho=0$$
$$\tilde{X}^{(n)}(X^{(n)}\rho-\hat{x}u)=-\lambda v\Rightarrow\tilde{\rho}\tilde{X}^{(n)}(X^{(n)}-\hat{x}u\tilde{v})\rho=-\lambda\tilde{\rho}v=-\lambda;$$

d'où: $$\|X^{(n)}\rho-\hat{x}\|^2=-\lambda=\det\left\{\begin{pmatrix} \tilde{u} \\ \tilde{X}^{(n)} \end{pmatrix}(u,X^{(n)})\right\}/\det\left\{\begin{pmatrix} \tilde{u} \\ \tilde{\Delta X}^{(n)} \end{pmatrix}(u,\Delta X^{(n)})\right\}.$$

Pour achever la démonstration, il suffit de noter que l'unicité du point stationnaire implique son identité avec la solution unique du problème de minimisation (P_1).

3. Conservation de la limite par le procédé $C_1^{(m)}$

3.1 Le procédé $C_1^{(m)}$.

Nous nous limiterons désormais au cas où k=1. Le procédé s'écrit alors:

$$C_1^{(m)}(x_n)=\frac{\begin{vmatrix} \sum_{i=0}^{m-1}x_{n+i} & x_{n+m}-x_n \\ \sum_{i=0}^{m-1}\Delta\bar{x}_{n+i}x_{n+i} & \sum_{i=0}^{m-1}|\Delta x_{n+i}|^2 \end{vmatrix}}{\begin{vmatrix} m & x_{n+m}-x_n \\ \bar{x}_{n+m}-\bar{x}_n & \sum_{i=0}^{m-1}|\Delta x_{n+i}|^2 \end{vmatrix}}$$

C'est une généralisation du classique procédé δ^2 qu'on retrouve en choisissant m=2. Nous nous proposons maintenant de déterminer comment les propriétés de cette transformation s'étendent au procédé $C_1^{(m)}$.

Lubkin [7] et Marx [8] ont montré que le procédé δ^2 n'était pas régulier en exhibant une suite convergente dont le transformé par le procédé δ^2 ne converge pas. Toutefois, Lubkin [7] a montré que, si la suite transformée converge encore, alors la limite des deux suites est la même. C'est ce

résultat essentiel que nous nous proposons d'étendre aux procédés $C_1^{(m)}$. Toutefois, bien que Tucker [13] ait montré que, pour le procédé δ^2, ce résultat était encore valable dans le cas de suites de nombres complexes, nous nous limitons ici au cas de suites de nombres réels. Nous nous appuyerons pour cela sur le lemme suivant:

Lemme: *Si une suite de nombres réels $(x_n)_{n\geq 0}$ est telle qu'il existe un nombre positif M vérifiant:* $|\Delta^2 x_n| < M(\Delta x_n)^2$, *alors cette suite diverge.*

Démonstration: L'inégalité $|\Delta^2 x_n| < M(\Delta x_n)^2$ s'écrit encore: $-M(\Delta x_n)^2 < \Delta x_{n+1} - \Delta x_n < M(\Delta x_n)^2$, d'où: $(1-M\Delta x_n)\Delta x_n < \Delta x_{n+1} < (1+M\Delta x_n)\Delta x_n$. Supposons la convergence de la suite (x_n). Ceci assure: $\lim_{n\to\infty} \Delta x_n = 0$, et à tout $\varepsilon > 0$, on peut associer $n_\varepsilon \in \mathbb{N}$ tel que: $n \geq n_\varepsilon \Rightarrow |M\Delta x_n| < \varepsilon$.

Alors $n \geq n_\varepsilon$ assure $1-M\Delta x_n > 0$ et $1+M\Delta x_n > 0$, d'où $\text{signe}(\Delta x_{n+1}) = \text{signe}(\Delta x_n)$, $\forall n \geq n_1$, soit, par récurrence: $\text{signe}(\Delta x_n) = \text{signe}(\Delta x_{n_1})$, $\forall n \geq n_1$.

Supposons que $\text{signe}(\Delta x_{n_1}) > 0$. Nous avons alors la double inégalité:

$$\frac{1}{\Delta x_n(1+M\ \Delta x_n)} < \frac{1}{\Delta x_{n+1}} < \frac{1}{\Delta x_n(1-M\ \Delta x_n)}, \quad \forall n \geq n_1 ;$$

Ceci équivaut à: $$\frac{-M}{1+M\ \Delta x_n} < \frac{1}{\Delta x_{n+1}} - \frac{1}{\Delta x_n} < \frac{M}{1-M\ \Delta x_n}$$

Fixons $\varepsilon \in]0,1[$. Pour tout $n \geq n_\varepsilon$, nous aurons: $\frac{1}{\Delta x_{n+1}} - \frac{1}{\Delta x_n} < \frac{M}{1-\varepsilon}$, d'où:

$$\frac{1}{\Delta x_n} < \frac{1}{\Delta x_{n_\varepsilon}} + \frac{(n-n_\varepsilon)M}{1-\varepsilon} \text{ pour tout entier } n > n_\varepsilon, \text{ soit: } \Delta x_n > \frac{1}{\frac{M}{1-\varepsilon}(n-n_\varepsilon)+\frac{1}{\Delta x_{n_\varepsilon}}}.$$

Cette minoration est strictement positive dès que: $n > n_\varepsilon - \frac{1-\varepsilon}{M\ \Delta x_{n_\varepsilon}}$.

Alors la divergence de la série de terme général $u_n = \frac{1}{an+b}$ garantit celle de la suite (x_n).

Dans le cas où $\text{signe}(\Delta x_n) < 0$, considérons la suite de terme général $y_n = -x_n$. Le raisonnement précédent établit alors la divergence de la suite y_n, d' où celle de la suite x_n.

Proposition 5. *Si le procédé $C_1^{(m)}$ transforme une suite convergente de nombres réels en une suite convergente, alors ces deux suites ont la même limite.*

Démonstration: Soit x_n le terme général de la suite initiale, $y_n = C_1^{(m)}(x_n)$ celui de la suite transformée et nous notons $\hat{x}$ et $\hat{y}$ les limites respectives de ces deux suites qu'on suppose convergentes.

Si nous introduisons les trois vecteurs de $\mathbb{R}^m$ suivants: $X_n = \begin{pmatrix} x_n \\ \vdots \\ x_{n+m-1} \end{pmatrix}$, $U_n = \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}$ et $\Delta X_n = \begin{pmatrix} \Delta x_n \\ \vdots \\ \Delta x_{n+m-1} \end{pmatrix}$,

Annexe 1: transformation de suites par approximation.

nous aurons: $y_n = \dfrac{\begin{vmatrix} (U,X_n) & (U,\Delta X_n) \\ (\Delta X_n,X_n) & (\Delta X_n,\Delta X_n) \end{vmatrix}}{\begin{vmatrix} (U,U) & (U,\Delta X_n) \\ (\Delta X_n,U) & (\Delta X_n,\Delta X_n) \end{vmatrix}}$, où (X,Y) représente le produit scalaire euclidien de $\mathbb{R}^m$: $(X,Y)=\sum_{i=0}^{m-1} x_{n+i}y_{n+i}$.

Si nous posons, $e_n = x_n - \hat{x}$ et $E_n = \begin{pmatrix} e_n \\ \vdots \\ e_{n+m-1} \end{pmatrix}$, l'écart $y_n - \hat{y}$ s'écrit: $y_n - \hat{y} = \dfrac{\begin{vmatrix} (U,E_n) & (U,\Delta E_n) \\ (\Delta E_n,E_n) & (\Delta E_n,\Delta E_n) \end{vmatrix}}{\begin{vmatrix} (U,U) & (U,\Delta E_n) \\ (\Delta E_n,U) & (\Delta E_n,\Delta E_n) \end{vmatrix}}$.

Notons que l'éventuelle nullité de ΔE_n interdit la mise en œuvre du procédé $C_1^{(m)}$. Puisque y_n tend vers $\hat{y}$ quand n tend vers l'infini, il existe un indice n_0 au delà duquel on a: $\|\Delta E_n\| \neq 0$.

Posons alors: $Q_n = \Delta E_n / \|\Delta E_n\|$, $\forall n \geq n_0$, où $\|\ .\ \|$ représente la norme euclidienne: $\|X\| = (X,X)^{1/2}$.

On obtient alors: $y_n - \hat{x} = \dfrac{\begin{vmatrix} (U,E_n) & (U,Q_n) \\ (Q_n,E_n) & (Q_n,Q_n) \end{vmatrix}}{\begin{vmatrix} (U,U) & (U,Q_n) \\ (Q_n,U) & (Q_n,Q_n) \end{vmatrix}}$.

Ce quotient de déterminants tend vers $\hat{y} - \hat{x} \neq 0$. Puisque $\lim_{n\to\infty} x_n = \hat{x}$ implique $\lim_{n\to\infty} E_n = 0$, le déterminant du numérateur tend vers 0. Celui du dénominateur tendra donc lui aussi vers 0. Si nous notons $V = U/\|U\|$ le vecteur unitaire de $\mathbb{R}^m$ dont toutes les composantes sont égales à $m^{-1/2}$, nous avons: $\lim_{n\to\infty} (\|Q_n\|^2 - (Q_n,V)^2) = 0$, avec $\|Q_n\| = \|V\| = 1$. Ceci implique: $\lim_{n\to\infty} |(Q_n,V)| = 1$; la suite (Q_n) a donc au plus deux points d'accumulation, les vecteurs V et -V, ce qui entraîne: $\lim_{n\to\infty} \dfrac{\Delta e_{n+1}}{\Delta e_n} = 1$, d'où:

$$\lim_{n\to\infty} \frac{\Delta^2 e_n}{\Delta e_n} = 0.$$

Considérons maintenant l'expression: $y_n - x_n = y_n - \hat{x} - e_n = \dfrac{\begin{vmatrix} (U,E_n - e_nU) & (U,\Delta E_n) \\ (\Delta E_n,E_n - e_nU) & (\Delta E_n,\Delta E_n) \end{vmatrix}}{\begin{vmatrix} (U,U) & (U,\Delta E_n) \\ (\Delta E_n,U) & (\Delta E_n,\Delta E_n) \end{vmatrix}}$.

Remarquons que le dénominateur peut s'écrire sous la forme:

$$D_n = \begin{vmatrix} (U,U) & (U,\Delta E_n) \\ (\Delta E_n,U) & (\Delta E_n,\Delta E_n) \end{vmatrix} = \begin{vmatrix} (U,U) & (U,\Delta E_n - \Delta e_nU) \\ (\Delta E_n - \Delta e_nU,U) & (\Delta E_n - \Delta e_nU,\Delta E_n - \Delta e_nU) \end{vmatrix},$$

ce qui nous permet d'affirmer que $\Delta E_n - \Delta e_n U \neq 0$ au delà d'un certain indice n_0.
De même, le numérateur s'écrira:

$$N_n = \begin{vmatrix} (U,E_n-e_nU) & (U,\Delta E_n) \\ (\Delta E_n,E_n-e_nU) & (\Delta E_n,\Delta E_n) \end{vmatrix} = \begin{vmatrix} (U,E_n-e_nU) & (U,\Delta E_n) \\ (\Delta E_n-\Delta e_nU,E_n-e_nU) & (\Delta E_n-\Delta e_nU,\Delta E_n) \end{vmatrix}.$$

Puisque $\Delta E_n - \Delta e_n U$ n'est pas nul, il est possible de diviser numérateur et dénominateur par $\|\Delta E_n - \Delta e_n U\|^2$ dès que $n \geq n_0$. Il vient alors: $y_n - \hat{x} - e_n = \frac{N'_n}{D'_n}$ avec:

$$D'_n = \frac{1}{\|\Delta E_n - \Delta e_n U\|^2} \begin{vmatrix} (U,U) & (U,\Delta E_n-\Delta e_nU) \\ (\Delta E_n-\Delta e_nU,U) & (\Delta E_n-\Delta e_nU,\Delta E_n-\Delta e_nU) \end{vmatrix}$$

et:

$$N'_n = \frac{1}{\|\Delta E_n - \Delta e_n U\|^2} \begin{vmatrix} (U,E_n-e_nU) & (U,\Delta E_n) \\ (\Delta E_n-\Delta e_nU,E_n-e_nU) & (\Delta E_n-\Delta e_nU,\Delta E_n) \end{vmatrix}.$$

Supposons que 0 soit point d'accumulation de la suite D'_n. Alors il existe un sous-ensemble infini d'indices N' de $\mathbb{N}$ tel que $\lim_{\substack{n\to\infty \\ n\in N'}} D'_n = 0$ et ceci implique que la suite des vecteurs $R_n = \frac{\Delta E_n - \Delta e_n U}{\|\Delta E_n - \Delta e_n U\|}$ a au plus deux points d'accumulation, les vecteurs V et -V, ce qui est impossible car la première composante de R_n est toujours nulle alors que celle de V vaut $m^{-1/2}$. Au delà d'un certain rang n_0, D'_n est donc non nul. D'autre part $D'_n \geq 0$ grace à l'inégalité de Schwarz. Il existe donc $\rho > 0$ tel que $D'_n > \rho$, $\forall n > n_0$.

Puisque $y_n - \hat{x} - e_n$ tend vers $\hat{y} - \hat{x} \neq 0$ quand n tend vers l'infini, le numérateur N'_n n'admet pas 0 comme point d'accumulation. Plus précisément, il existe n_1 tel que pour tout $n \geq n_1$, on ait:

$$|N'_n| > \rho\, |\hat{y} - \hat{x}|\, /2.$$

Posons maintenant: $F_n = E_n - e_n U = \begin{pmatrix} 0 \\ \Delta e_n \\ \Delta e_n + \Delta e_{n+1} \\ \vdots \\ \sum_{i=0}^{m-2} \Delta e_{n+i} \end{pmatrix}$, d'où : $\Delta F_n = \Delta E_n - \Delta e_n U = \begin{pmatrix} 0 \\ \Delta^2 e_n \\ \Delta^2 e_n + \Delta^2 e_{n+1} \\ \vdots \\ \sum_{i=0}^{m-2} \Delta^2 e_{n+i} \end{pmatrix}$.

La numérotation précédente se traduit par:

$$\left| \frac{1}{\|\Delta F_n\|} \begin{vmatrix} (U,F_n) & (U,\Delta E_n) \\ \left(\frac{\Delta F_n}{\|\Delta F_n\|}, F_n\right) & \left(\frac{\Delta F_n}{\|\Delta F_n\|}, \Delta E_n\right) \end{vmatrix} \right| > |\hat{y} - \hat{x}|\, /2$$

Le déterminant précédent est certainement majoré par $2\,\|U\|\,\|\Delta E_n\|\,\|F_n\|$, d'où:

$$\|\Delta F_n\| < \frac{4\ \|U\|\ \|\Delta E_n\|\ \|F_n\|}{\rho\ |\hat{y}-\hat{x}\ |}\ .$$

Puisque $\lim_{n\to\infty} \frac{\Delta e_{n+1}}{\Delta e_n} = 1$, nous avons: $\lim_{n\to\infty} \frac{F_n}{\Delta e_n} = \begin{pmatrix} 0 \\ 1 \\ 2 \\ \vdots \\ n-1 \end{pmatrix}$ et $\lim_{n\to\infty} \frac{E_n}{\Delta e_n} = \begin{pmatrix} 1 \\ 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix}$

Donc: $\lim_{n\to\infty} \frac{\|F_n\|^2}{(\Delta e_n)^2} = \sum_{i=0}^{m-1} i^2 = m(m-1)(2m-1)/6$ et $\lim_{n\to\infty} \frac{\|E_n\|^2}{(\Delta e_n)^2} = \|U\|^2 = m$,

d'où: $\lim_{n\to\infty} \frac{2\ \|U\|\ \|\Delta E_n\|\ \|F_n\|}{(\Delta e_n)^2} = m(2m(m-1)(2m-1)/3)^{1/2}$.

D'autre part, $|\Delta^2 x_n| \le \|\Delta F_n\|$. Il suffit alors de choisir: $K = \frac{m(2m(m-1)(2m-1)/3)^{1/2}}{\rho\ |\ \hat{y} - \hat{x}\ |}$

pour avoir: $|\Delta^2 x_n| \le K(\Delta x_n)^2$, et le lemme précédent permet de conclure.

4 Accélération de la convergence par le procédé $C_1^{(m)}$.

4.1 Suites à convergence géométrique.

Dans ce paragraphe comme dans celui qui suit, nous nous intéressons plus précisément à l'action des procédés $C_1^{(m)}$ sur une classe de suites que nous définissons maintenant.

Définition 1 *Une suite (x_n) de nombres complexes sera dite ...*

*... **géométrique** de raison $\lambda \in \mathbb{C}\setminus(0,1)$ et de centre $\hat{x} \in \mathbb{C}$ si elle vérifie:*

$$x_n - \hat{x} = \lambda^n (x_0 - \hat{x}),\ \forall n \ge 0.$$

*... **à comportement géométrique** de raison $\lambda \in \mathbb{C}\setminus(0,1)$ et de centre $\hat{x} \in \mathbb{C}$ si elle vérifie:*

$$\lim_{n\to\infty} \frac{x_{n+1} - \hat{x}}{x_n - \hat{x}} = \lambda .$$

*... **à convergence géométrique** (ou **linéaire**) de raison λ et de limite $\hat{x}$ si elle a un comportement géométrique dont la raison λ a un module inférieur à l'unité.*

Dans le cas où $|\lambda|>1$, cette suite diverge, ou plutôt elle converge vers le point à l'infini qui compactifie $\mathbb{C}$ et $\hat{x}$ est son antilimite [11]. Dans le cas où $|\lambda|=1$, elle peut aussi bien converger que diverger.

Définition 2: *Une transformation de suites T **accélère la convergence** d'une suite (x_n) de limite $\hat{x}$ si l'on a:* $\lim_{n\to\infty} \frac{T(x_n) - \hat{x}}{x_n - \hat{x}} = 0.$

4.2 Accélération de la convergence géométrique.

Proposition 6. *Si la suite (x_n) est à convergence géométrique de raison λ (donc $|\lambda|\leq 1$), alors le procédé $C_1^{(m)}$ accélère la convergence de cette suite.*

Démonstration: Par définition, la suite satisfait une équation de récurrence de la forme:

$$x_{n+1}-\hat{x} = \lambda(x_n-\hat{x})(1-\varepsilon_n),\ \forall n\geq 0\ ,\ \text{où}\ \hat{x}\in\mathbb{C},\ \lambda\in\mathbb{C}\setminus\{0,1\}\ \text{et}\ \lim_{n\to\infty}\varepsilon_n=0.$$

La relation: $x_{n+1}-\hat{x} = \lambda^1(x_n-\hat{x})\prod_{j=0}^{1-1}(1-\varepsilon_{n+j})$ entraîne donc, pour p fixé: $\lim_{n\to\infty}\dfrac{x_{n+p}-\hat{x}}{x_n-\hat{x}} = \lambda^p$. Il

s'ensuit que, si nous montrons que: $\lim_{n\to\infty}\dfrac{C_1^{(m)}(x_n)-\hat{x}}{x_n-\hat{x}}=0$, nous aurons aussi: $\lim_{n\to\infty}\dfrac{C_1^{(m)}(x_n)-\hat{x}}{x_{n+m}-\hat{x}}=0$.

Compte tenu de l'hypothèse, nous pouvons expliciter les quatre expressions suivantes:

1° $x_{n+m}-x_n=[\lambda^m\prod_{j=0}^{m-1}(1+\lambda_{n+j})-1]\ (x_n-\hat{x})$, d'où : $\lim_{n\to\infty}\dfrac{x_{n+m}-x_n}{x_n-\hat{x}} = \lambda^m-1.$

2° $\Delta x_{n+i}=[|(1+\varepsilon_{i+n})-1]\ \lambda^i\prod_{j=0}^{i-1}(1+\varepsilon_{n+j})(x_n-\hat{x})$, d'où : $\lim_{n\to\infty}\dfrac{\Delta x_{n+i}}{x_n-\hat{x}} = (\lambda-1)\lambda^i$, et

$$\lim_{n\to\infty}\frac{\sum_{i=0}^{m-1}|\Delta x_{n+i}|^2}{|x_n-\hat{x}|^2} = |\lambda-1|^2\sum_{j=0}^{m-1}|\lambda|^{2j} = \frac{|1-\lambda|^2(1-|\lambda|^{2m})}{1-|\lambda|^2}$$

3° $\sum_{i=0}^{m-1}(x_{n+i}-\hat{x})=(x_n-\hat{x})\sum_{i=0}^{m-1}[\lambda^i\prod_{j=0}^{i-1}(1+\varepsilon_{n+j})]$, d'où : $\lim_{n\to\infty}\dfrac{\sum_{i=0}^{m-1}x_{n+i}-\hat{x}}{x_n-\hat{x}} = \dfrac{1-\lambda^m}{1-\lambda}.$

4° $\sum_{i=0}^{m-1}[(x_{n+i}-\hat{x})\Delta\bar{x}_{n+i}]=(x_n-\hat{x})(\bar{x}_n-\hat{x})\sum_{i=0}^{m-1}[\lambda^i\prod_{j=0}^{i-1}(1+\varepsilon_{n+j})(\bar{\lambda}(1+\bar{\varepsilon}_{n+i})-1)\bar{\lambda}^i\prod_{j=0}^{i-1}(1+\bar{\varepsilon}_{n+j})]$,

d'où : $\lim_{n\to\infty}\dfrac{\sum_{i=0}^{m-1}[(x_{n+i}-\hat{x})\ \Delta\bar{x}_{n+i}]}{|x_n-\hat{x}|^2} = \sum_{j=0}^{m-1}(\bar{\lambda}-1)\ |\lambda|^{2j} = \dfrac{1-|\lambda|^{2m}}{1-|\lambda|^2}(\bar{\lambda}-1)$.

Il s'ensuit qu'il existe des nombres $\nu_n, \nu'_n, \eta_n, \eta'_n$ tendant vers zéro quand n tend vers l'infini et tels

que le nombre $Z_1(m,n)$ défini par: $Z_1(m,n) = \dfrac{1}{|x_n-\hat{x}|^3}\begin{vmatrix} \sum_{i=0}^{m-1}(x_{n+i}-\hat{x}) & x_{n+m}-x_n \\ \sum_{i=0}^{m-1}[(x_{n+i}-\hat{x})\Delta\bar{x}_{n+i}] & \sum_{i=0}^{m-1}|\Delta x_{n+i}|^2 \end{vmatrix}$

s'écrive encore: $$Z_1(m,n) = \begin{vmatrix} \dfrac{1-\lambda^m}{1-\lambda}(1+\nu_n) & (\lambda^m-1)(1+\eta_n) \\ \dfrac{1-|\lambda|^{2m}}{1-|\lambda|^2}\,\bar{\lambda}-1)\,(1+\eta'_n) & \dfrac{|1-\lambda|^2(1-|\lambda|^{2m})}{1-|\lambda|^2}(1+\nu'_n) \end{vmatrix}.$$

Des opérations élémentaires donnent alors:

$$Z_1(m,n)=\frac{(1-\bar{\lambda})(1-\lambda^m)(1-|\lambda|^{2m})}{1-|\lambda|^2}\begin{vmatrix} 1+\nu_n & -1-\eta_n \\ -1-\nu'_n & 1+\eta'_n \end{vmatrix}=\frac{(1-\bar{\lambda})(1-\lambda^m)(1-|\lambda|^{2m})}{1-|\lambda|^2}\begin{vmatrix} 1+\nu_n & \eta_n-\nu_n \\ 1+\nu'_n & \eta'_n-\nu'_n \end{vmatrix}.$$

D'où: $$\lim_{n\to\infty}\left(\frac{1}{|x_n-\hat{x}|^3}\begin{vmatrix} \sum_{i=0}^{m-1}(x_{n+i}-\hat{x}) & x_{n+m}-x_n \\ \sum_{i=0}^{m-1}[(x_{n+i}-\hat{x})\Delta\bar{x}_{n+i}] & \sum_{i=0}^{m-1}|\Delta x_{n+i}|^2 \end{vmatrix}\right)=0.$$

De même, il existe de nombres ν_n et ν'_n tels que le nombre $Z_2(m,n)$ défini par:

$$Z_2(m,n)=\frac{1}{|x_n-\hat{x}|^2}\begin{vmatrix} m & x_{n+m}-x_n \\ \bar{x}_{n+m}-\bar{x}_n & \sum_{i=0}^{m-1}|\Delta x_{n+i}|^2 \end{vmatrix}$$

s'écrive encore:

$$Z_2(m,n)=\begin{vmatrix} m & (\lambda^m-1)(1+\nu_n) \\ (\bar{\lambda}^m-1)(1+\bar{\nu}_n) & \dfrac{|1-\lambda|^2(1-|\lambda|^{2m})}{1-|\lambda|^2}(1+\nu'_n) \end{vmatrix}=|1-\lambda|^2\begin{vmatrix} m & \dfrac{1-\lambda^m}{1-\lambda}(1+\nu'_n) \\ \dfrac{1-\bar{\lambda}^m}{1-\bar{\lambda}}(1+\nu'_n) & \dfrac{1-|\lambda|^{2m}}{1-|\lambda|^2}(1+\nu'_n) \end{vmatrix}.$$

Quand n tend vers l'infini, cette expression tend vers: $$\lim_{n\to\infty} Z_2(m,n) = |1-\lambda|^2\begin{vmatrix} m & \dfrac{1-\lambda^m}{1-\lambda} \\ \dfrac{1-\bar{\lambda}^m}{1-\bar{\lambda}} & \dfrac{1-|\lambda|^{2m}}{1-|\lambda|^2} \end{vmatrix}.$$

Ce déterminant n'est pas nul, car il représente l'expression : $\|U\|^2\|L\|^2-(U,L)^2$,

où U et L sont les deux vecteurs de $\mathbb{C}^m$ définis par: $U=\begin{pmatrix}1\\1\\\vdots\\1\end{pmatrix}$ et $L=\begin{pmatrix}1\\\lambda\\\vdots\\\lambda^{m-1}\end{pmatrix}$

et cette expression est strictement positive dès que $|\lambda|<1$ (inégalité de Schwartz).

5. Note sur le conditionnement du procédé $C_1^{(m)}$.

5.0 Présentation.

Lors de l'introduction des procédés $C_k^{(m)}$ reposant sur l'approximation, notre objet était d'aboutir à une transformation qui soit moins sensible à l'imprécision des valeurs de la suite initiale que les

transformations classiques reposant sur l'approximation. Nous nous proposons ici de vérifier que les procédés en question jouissent bien de cette propriété. Toutefois, en raison de la complexité des calculs auxquels conduit une telle étude, nous nous sommes contentés de l'aborder dans le cas où k=1. Cette question de la sensibilité de la solution d'un problème aux perturbations dont sont entachées les données est en fait celle de la détermination du conditionnement du problème. Aussi notre premier souci sera-t-il de préciser quelques notions relatives au problème particulier qui nous intéresse ici en introduisant un nombre de conditionnement "ponctuel" et un nombre de conditionnement "global".

Après avoir établi un lemme permettant d'expliciter les dérivées partielles de la transformée $C_k^{(m)}(x_n)$ en fonction des valeurs de la suite initiale, nous déterminons un nombre de conditionnement ponctuel pour le procédé $C_1^{(m)}$. L'expression de ce nombre de conditionnement (que nous aurions pu exhiber dans le cas général des procédés $C_k^{(m)}$) ne paraît guère intéressante en raison du coût opératoire de son calcul. Toutefois, si nous nous limitons aux suites géométriques, c'est à dire aux suites que le procédé δ^2 transforme en suites constantes, nous pouvons alors déterminer le nombre de conditionnement global du procédé $C_1^{(m)}$ vis à vis d'une telle suite. En particulier, on peut montrer que ce nombre tend vers 1 quand on fait tendre m vers l'infini, ce qui signifie que, pour pour une suite géométrique perturbée, il existe toujours un entier m tel que le procédé $C_1^{(m)}$ amplifiera aussi peu que l'on veut les perturbations de la suite initiale.

5.1 Conditionnement d'une transformation vis à vis d'une suite.

Donnons-nous une suite $(x_n)_{n\geq 0}$ de nombres réels et une transformation T_m dont la mise en œuvre nécessite m+1 valeurs consécutives de la suite: $T_m : x_n \rightarrow y_n = T_m(x_n)$, $\forall n \geq 0$, et notons U l'application de $\mathbb{R}^{m+1}$ dans $\mathbb{R}$ associée à la transformation de suite T_m : $T_m(x_n) = U(x_n,\ldots,x_{n+m})$. A tout $\delta \geq 0$, on associe la boule $B_m(x_n,\delta)$ des suites finies $(x'_n,\ldots,x'_{n+m})$ vérifiant : $|x'_i - x_i| \leq \delta$ pour $i=n,\ldots,n+m$, on note Ω_i la boule de centre x_i et de rayon δ (pour $i=n,\ldots,n+m$), et on pose:

$$U(\Omega_n,\ldots,\Omega_{n+m}) = \{\, z \mid z = U(x'_n,\ldots,x'_{n+m}),\ x'_i \in \Omega_i, i=n,\ldots,n+m \,\}.$$

Suivant Rice [9], à tout nombre $\delta \geq 0$, nous pouvons associer le *δ-conditionnement ponctuel* de la transformation T_m vis à vis de la suite (x_n) pour l'indice p par:

$$\Gamma_\delta(T_m,(x_n),p) = \inf \{\, \rho \mid U(B_m(x_n,\delta)) \subset B_0(T_m(x_p),\rho\delta) \},$$

et s'il existe, son *conditionnement ponctuel limite*: $\Gamma_0(T_m,(x_n),p) = \lim_{\delta \to 0} \Gamma_\delta(T_m,(x_n),p)$.

Si nous supposons que la fonction U associée à la transformation T_m est continument différentiable par rapport à ses m-1 arguments au voisinage du point $(x_p,\ldots,x_{p+m})$, alors Γ_δ tend vers le conditionnement limite Γ_0 et nous avons: $\Gamma_0(T_m,(x_n),p) = \sum_{i=p}^{p+m} \left| \frac{\partial}{\partial x_i} T_m(x_p) \right|$.

Dans ces conditions, le nombre Γ_0 fournira une bonne approximation de Γ_δ, coefficient maximal d'amplification des erreurs de module inférieur à δ par le procédé T_m pour l'indice p.

Le nombre $\bar{\Gamma}_0(T_m,(x_n))= \overline{\lim_{p\to\infty}}\ \Gamma_0(T_m,(x_n),p)$ est borné dès que les $\Gamma_0(T_m,(x_n),p)$ sont uniformément bornés. On l'appellera *conditionnement global* du procédé T_m par rapport à la suite (x_n).

5.2 lemme. *Soient N(u) et D(u) deux déterminants d'ordre k+1 ne différant que par leur première colonne et dont les (k+1)(k+2) éléments sont des fonctions dérivables de la variable u:*

$$N(u)=\begin{vmatrix} b_0 & a_{01} & \cdots & a_{0k} \\ b_1 & a_{11} & \cdots & a_{1k} \\ \vdots & \vdots & & \vdots \\ b_k & a_{k1} & \cdots & a_{kk} \end{vmatrix} \qquad D(u)=\begin{vmatrix} a_{00} & a_{01} & \cdots & a_{0k} \\ a_{10} & a_{11} & \cdots & a_{1k} \\ \vdots & \vdots & & \vdots \\ a_{k0} & a_{k1} & \cdots & a_{kk} \end{vmatrix}$$

Pour toute valeur de u n'annulant pas D(u), la fonction f(u)=N(u)/D(u) est dérivable et sa dérivée

s'écrit: $f'(u)=M(u)/(D(u))^2$ *avec:* $M(u)=\begin{vmatrix} \beta & \alpha_0 & \cdots & \alpha_k \\ b_0 & a_{00} & \cdots & a_{0k} \\ \vdots & \vdots & & \vdots \\ b_k & a_{k0} & \cdots & a_{kk} \end{vmatrix}$

où $\beta=\begin{vmatrix} b'_0 & a_{01} & \cdots & a_{0k} \\ b'_1 & a_{11} & \cdots & a_{1k} \\ \vdots & \vdots & & \vdots \\ b'_k & a_{k1} & \cdots & a_{kk} \end{vmatrix}$ *et* $\alpha_l=\begin{vmatrix} a'_{01} & a_{01} & \cdots & a_{0k} \\ a'_{11} & a_{11} & \cdots & a_{1k} \\ \vdots & \vdots & & \vdots \\ a'_{k1} & a_{k1} & \cdots & a_{kk} \end{vmatrix}$ *pour* $l=0,\ldots,k.$

Démonstration: La démonstration (assez technique) repose sur l'identité de Sylvester [4]. Nous avons:

$$f'(u)=\frac{N'(u)D(u)-D'(u)N(u)}{(D(u))^2},\ \text{avec}\ N'(u)=\sum_{l=0}^{k}N'_l(u)\ \text{et}\ D'(u)=\sum_{l=0}^{k}D'_l(u),$$

où $N'_l(u)$ est un déterminant d'ordre k+1 obtenu en remplaçant dans N(u) chaque terme de la i-ème ligne par sa dérivée et où $D'_l(u)$ est construit de la même façon à partir de D(u).

Si nous notons $\tilde{N}'_l(u)=(-1)^l N'_l(u)$ et $\tilde{D}'_l(u)=(-1)^l D'_l(u)$ les déterminants obtenus en faisant glisser la ligne d'indice l en première position, les lignes d'indice o,...,l-1 étant décallées d'un rang vers le bas, nous aurons: $f'(u)=M(u)/(D(u))^2$, avec:

$$M(u)=\sum_{l=0}^{k}(-1)^l\left(\tilde{N}'_l(u)D(u)-\tilde{D}'_l(u)N(u)\right).$$

Grace à l'identité de Sylvester, le terme entre crochets s'écrit:

$$\begin{vmatrix} b'_1 & a'_{10} & \dots & a'_{1k} \\ b_0 & a_{00} & \dots & a_{0k} \\ \vdots & \vdots & & \vdots \\ b_k & a_{k0} & \dots & a_{kk} \end{vmatrix} * \Delta_1,$$

où Δ_1 est le mineur relatif à a_{01} dans D(u) ou à b_1 dans N(u). Il s'ensuit que:

$$M(u)= \begin{vmatrix} \sum_{l=0}^{k}(1)^l b'_1 \Delta_1 & \sum_{l=0}^{k}(-1)^l a'_{10}\Delta_1 & \dots & \sum_{l=0}^{k}(-1)^l a'_{1k}\Delta_1 \\ b_0 & a_{00} & \dots & a_{0k} \\ \vdots & \vdots & & \vdots \\ b_k & a_{k0} & \dots & a_{kk} \end{vmatrix}$$

et il suffit de noter que chaque terme de la première ligne est le développement d'un des déterminants β ou α_i par rapport à sa première colonne pour aboutir au résultat annoncé.

5.3 Expression de $\frac{\partial}{\partial x_p}(C_1^{(m)}(x_n))$.

Nous pouvons écrire $C_1^{(m)}(x_0)$ sous la forme:

$$C_1^{(m)}(x_0)= \begin{vmatrix} b_0 & a_{01} & \dots & a_{0k} \\ b_1 & a_{11} & \dots & a_{1k} \\ \vdots & \vdots & & \vdots \\ b_k & a_{k1} & \dots & a_{kk} \end{vmatrix} / \begin{vmatrix} a_{00} & a_{01} & \dots & a_{0k} \\ a_{10} & a_{11} & \dots & a_{1k} \\ \vdots & \vdots & & \vdots \\ a_{k0} & a_{k1} & \dots & a_{kk} \end{vmatrix}$$

avec: $b_0=\sum_{l=0}^{m-k} x_l$, $\quad b_i=\sum_{l=0}^{m-k} x_l \Delta x_{l+i-1}$ (pour $i=1,\dots,k$),

$a_{00}=m-k+1$, $\quad a_{0i}=a_{i0}=x_{m-k+i}-x_{i-1}$ (pour $i=1,\dots,k$),

$a_{ij}=a_{ji}=\sum_{l=0}^{m-k}\Delta x_{l+i-1}\Delta x_{l+i-1}$ (pour $i,j=1,\dots,k$).

En explicitant les dérivées partielles de ces termes relativement à x_p pour $p=0,\dots,m$ et en les reportant dans l'expression de la dérivée de $C_k^{(m)}(x_0)$ par rapport à un paramètre fournie par le lemme précédent, on peut ainsi obtenir les dérivées partielles de la transformation $C_k^{(m)}(x_n)$ la plus générale et, par suite, le conditionnement ponctuel de la transformation.

L'expression de ce nombre de conditionnement ponctuel n'est certes pas totalement dépourvue d'intérêt puisqu'elle permet, pour une suite numérique donnée, d'avoir des informations sur la stabilité de la suite transformée par rapport à la précision de la suite initiale. Toutefois, sa

complexité nous interdit d'en tirer des résultats théoriques assez généraux et le coût opératoire de son évaluation amoindrissent notablement son intérêt pratique.

Par contre, si nous nous limitons au cas où k=1, le calcul est praticable, quoiqu'encore laborieux, et il conduit à des résultats tangibles lorsqu'on se limite à une certaine classe de suites. Par la suite, nous nous limiterons donc au cas où k=1.

Proposition 7. *Pour p=0,...,m, nous avons:* $\dfrac{\partial}{\partial x_{n+p}}(C_k^{(m)}(x_n)) = \dfrac{M_p^{(n)}}{(D^{(n)})^2}$ *avec:*

$$D=\begin{vmatrix} m & x_{n+m}-x_n \\ x_{n+m}-x_n & \Sigma(\Delta x_l)^2 \end{vmatrix},$$

$$M_0^{(n)} =\Sigma\Delta x_l(\Delta x_l-\Delta x_n)\begin{vmatrix} m & \Sigma x_{l+1} \\ \Sigma\Delta x_l & \Sigma x_{l+1}\Delta x_l \end{vmatrix} -(x_{n+m}-x_n)\begin{vmatrix} \Sigma(x_l-x_n) & \Sigma(x_{l+1}-x_{n+1}) \\ \Sigma\Delta x_l(x_l-x_n) & \Sigma(\Delta x_l)(x_{l+1}-x_{n+1}) \end{vmatrix},$$

$$M_m^{(n)}=\Sigma\Delta x_l(\Delta x_{n+m-1}-\Delta x_l)\begin{vmatrix} m & \Sigma x_l \\ \Sigma\Delta x_l & \Sigma x_l\Delta x_l \end{vmatrix} +(x_{n+m}-x_n)\begin{vmatrix} \Sigma(x_{n+m-1}-x_l) & \Sigma(x_{n+m}-x_{l+1}) \\ \Sigma\Delta x_l(x_{n+m-1}-x_l) & \Sigma(\Delta x_l)(x_{n+m}-x_{l+1}) \end{vmatrix},$$

et, pour p=1,...,m-1 : $M_p^{(n)}=D\,\Sigma(\Delta x_l)^2 +(x_{n+m}-x_n)^2\Delta^2 x_{n+p-1}\begin{vmatrix} l & x_n+x_{n+m} \\ m & \Sigma(x_l+x_{l+1}) \end{vmatrix},$

le symbole de sommation Σu_l *signifiant* $\sum_{l=n}^{n+m-1} u_l$.

Démonstration. Il est clair qu'il suffit d'établir ce résultat pour n=0. Nous avons alors:

$C_1^{(m)}(x_0)=\frac{N}{D}$, avec: $N=\begin{vmatrix} b_0 & a_{01} \\ b_1 & a_{11} \end{vmatrix}$, et: $D=\begin{vmatrix} a_{00} & a_{01} \\ a_{10} & a_{11} \end{vmatrix}$,

où: $b_0=\Sigma x_j$, $b_1=\Sigma x_j\Delta x_j$, $a_{00}=m$, $a_{01}=a_{10}=x_m-x_0=\Sigma\Delta x_j$, $a_{11}=\Sigma(\Delta x_j)^2$,

le symbole Σu_j signifiant $\sum_{j=0}^{m-1} u_j$.

En appliquant le lemme précédent, on obtient: $\dfrac{\partial}{\partial x_i}(C_1^{(m)}(x_0))= \dfrac{M_i^{(0)}}{D^2}$, avec :

$$M_i^{(0)}=\begin{vmatrix} \begin{vmatrix} b_0^{(i)} & a_{01} \\ b_1^{(i)} & a_{11} \end{vmatrix} & \begin{vmatrix} a_{00}^{(i)} & a_{01} \\ a_{10}^{(i)} & a_{11} \end{vmatrix} & \begin{vmatrix} a_{01}^{(i)} & a_{01} \\ a_{11}^{(i)} & a_{11} \end{vmatrix} \\ b_0 & a_{00} & a_{01} \\ b_1 & a_{10} & a_{11} \end{vmatrix} \quad \text{où } u^{(i)}=\frac{\partial u}{\partial x_i}, \text{ pour } i=0,...,m.$$

Pour poursuivre ce calcul, il nous faut distinguer 3 cas: i=0, i=m et 0<i<m.

1^er cas: $\boxed{i=0}$.

Alors: $b_0^{(0)}=1$, $b_1^{(0)}=2\Delta x_0-x_1$, $a_{00}^{(0)}=0$, $a_{10}^{(0)}=a_{01}^{(0)}=-1$, $a_{11}^{(0)}=-2\Delta x_0$, d'où:

$$M_0^{(0)}=\begin{vmatrix} \begin{vmatrix} 1 & a_{01} \\ 2\Delta x_0-x_1 & a_{11} \end{vmatrix} & \begin{vmatrix} 0 & a_{01} \\ -1 & a_{11} \end{vmatrix} & \begin{vmatrix} -1 & a_{01} \\ -2\Delta x_0 & a_{11} \end{vmatrix} \\ b_0 & a_{00} & a_{01} \\ b_1 & a_{10} & a_{11} \end{vmatrix}$$

Par des combinaisons de colonnes convenables, on obtient:

$$M_0^{(0)}=\begin{vmatrix} \begin{vmatrix} 0 & a_{01} \\ 0 & a_{11} \end{vmatrix} & \begin{vmatrix} 0 & a_{01} \\ -1 & a_{11} \end{vmatrix} & \begin{vmatrix} -1 & a_{01} \\ -\Delta x_0 & a_{11} \end{vmatrix} \\ b_0+a_{01}-x_1a_{00} & a_{00} & a_{01}-\Delta x_0a_{00} \\ b_1+a_{11}-x_1a_{10} & a_{10} & a_{11}-\Delta x_0a_{10} \end{vmatrix}$$

En développant par rapport à la première ligne:

$$M_0^{(0)}=a_{01}\begin{vmatrix} a_{01}-\Delta x_0a_{00} & b_0+a_{01}-x_1a_{00} \\ a_{11}-\Delta x_0a_{10} & b_0+a_{11}-x_1a_{10} \end{vmatrix}+(a_{11}-\Delta x_0a_{01})\begin{vmatrix} a_{00} & b_0+a_{01}-x_1a_{00} \\ a_{10} & b_0+a_{11}-x_1a_{10} \end{vmatrix}$$

d'où après simplification:

$$M_0^{(0)}=a_{01}\begin{vmatrix} a_{01}-\Delta x_0a_{00} & b_0-x_0a_{00} \\ a_{11}-\Delta x_0a_{10} & b_0-x_0a_{10} \end{vmatrix}+(a_{11}-\Delta x_0a_{01})\begin{vmatrix} a_{00} & b_0+a_{01} \\ a_{10} & b_0+a_{11} \end{vmatrix}.$$

Il suffit alors d'expliciter b_0, b_1, a_{00}, a_{01}, a_{10} et a_{11} pour obtenir:

$$M_0^{(0)}=(x_m-x_0)\begin{vmatrix} \Sigma(\Delta x_j-\Delta x_0) & \Sigma(x_j-x_0) \\ \Sigma\Delta x_j(\Delta x_j-\Delta x_0) & \Sigma\Delta x_j(x_j-x_0) \end{vmatrix}+\Sigma\Delta x_j(\Delta x_j-\Delta x_0)\begin{vmatrix} m & \Sigma x_{j+1} \\ x_m-x_0 & \Sigma x_{j+1}\Delta x_j \end{vmatrix}$$

2^nd cas: $\boxed{0<i<m}$

On a alors:

$$b_0^{(i)}=\frac{\partial b_0}{\partial x_i}=1,\ b_1^{(i)}=\frac{\partial b_1}{\partial x_i}=\Delta^2x_{i-1},\ a_{00}^{(i)}=\frac{\partial a_{00}}{\partial x_i}=0,\ a_{01}^{(i)}=\frac{\partial a_{01}}{\partial x_i}=a_{10}^{(i)}=\frac{\partial a_{10}}{\partial x_i}=0,\ a_{11}^{(i)}=\frac{\partial a_{11}}{\partial x_i}=\Delta^2x_{i-1}.$$

En reportant ces valeurs dans l'expression de $M_i^{(0)}$:

$$M_i^{(0)}=\begin{vmatrix} \begin{vmatrix} 1 & a_{01} \\ \Delta^2 x_{i-1} & a_{11} \end{vmatrix} & \begin{vmatrix} 0 & a_{01} \\ 0 & a_{11} \end{vmatrix} & \begin{vmatrix} 0 & a_{01} \\ -2\Delta^2 x_{i-1} & a_{11} \end{vmatrix} \\ b_0 & a_{00} & a_{01} \\ b_1 & a_{10} & a_{11} \end{vmatrix} = \begin{vmatrix} \begin{vmatrix} 1 & a_{01} \\ 0 & a_{11} \end{vmatrix} & \begin{vmatrix} 0 & a_{01} \\ 0 & a_{11} \end{vmatrix} & \begin{vmatrix} 0 & a_{01} \\ -2\Delta^2 x_{i-1} & a_{11} \end{vmatrix} \\ b_0+a_{01}/2 & a_{00} & a_{01} \\ b_1+a_{11}/2 & a_{10} & a_{11} \end{vmatrix}$$

$$=a_{11}\begin{vmatrix} a_{00} & a_{01} \\ a_{10} & a_{11} \end{vmatrix}-2\Delta^2 x_{i-1}a_{01}\begin{vmatrix} a_{00} & b_0+a_{01}/2 \\ a_{10} & b_1+a_{11}/2 \end{vmatrix}.$$

Notons alors que:

$$b_0+a_{01}/2=\sum_{j=0}^{m-1}x_j+\frac{1}{2}\sum_{j=0}^{m-1}\Delta x_j=\frac{1}{2}\left(\sum_{j=0}^{m-1}(x_j+\Delta x_j)+\sum_{j=0}^{m-1}x_j\right)=\frac{1}{2}\sum_{j=0}^{m-1}(x_j+x_{j+1}),$$

$$b_1+a_{11}/2=\sum_{j=0}^{m-1}x_j\Delta x_j+\frac{1}{2}\sum_{j=0}^{m-1}(\Delta x_j)^2=\frac{1}{2}\sum_{j=0}^{m-1}(x_{j+1}-x_j)(x_j+x_{j+1})=\frac{1}{2}\sum_{j=0}^{m-1}((x_{j+1})^2-(x_j)^2)=\frac{1}{2}((x_m)^2-(x_0)^2),$$

d'où:

$$M_i^{(0)}=D_x\Sigma(\Delta x_j)^2-2\Delta^2 x_{i-1}(x_m-x_0)\begin{vmatrix} m & \frac{1}{2}\Sigma(x_j+x_{j+1}) \\ x_m-x_0 & \frac{1}{2}((x_m)^2-(x_0)^2) \end{vmatrix}$$

$$=D_x\Sigma(\Delta x_j)^2-\Delta^2 x_{i-1}(x_m-x_0)^2\begin{vmatrix} 1 & x_m+x_0 \\ m & \Sigma(x_j+x_{j+1}) \end{vmatrix}.$$

3ème cas: $\boxed{i=m}$

Alors: $b_0^{(m)}=,\ b_1^{(m)}=x_{m-1},\ a_{00}^{(m)}=0,\ a_{10}^{(m)}=a_{01}^{(0)}=1,\ a_{11}^{(m)}=2\Delta x_{m-1},$

d'où: $$M_m^{(0)}=\begin{vmatrix} \begin{vmatrix} 1 & a_{01} \\ x_{m-1} & a_{11} \end{vmatrix} & \begin{vmatrix} 0 & a_{01} \\ 1 & a_{11} \end{vmatrix} & \begin{vmatrix} 1 & a_{01} \\ 2\Delta x_{m-1} & a_{11} \end{vmatrix} \\ b_0 & a_{00} & a_{01} \\ b_1 & a_{10} & a_{11} \end{vmatrix}$$

Par des combinaisons de colonnes convenables, on obtient:

$$M_m^{(0)}=\begin{vmatrix} \begin{vmatrix} 0 & a_{01} \\ 0 & a_{11} \end{vmatrix} & \begin{vmatrix} 0 & a_{01} \\ 1 & a_{11} \end{vmatrix} & \begin{vmatrix} 1 & a_{01} \\ \Delta x_{m-1} & a_{11} \end{vmatrix} \\ b_0-x_{m-1}a_{00} & a_{00} & a_{01}-\Delta x_{m-1}a_{00} \\ b_1-x_{m-1}a_{10} & a_{10} & a_{11}-\Delta x_{m-1}a_{10} \end{vmatrix}$$

En développant par rapport à la première ligne:

$$M_m^{(0)} = a_{01}\begin{vmatrix} b_0-x_{m-1}a_{00} & a_{01}-\Delta x_{m-1}a_{00} \\ b_1-x_{m-1}a_{10} & a_{11}-\Delta x_{m-1}a_{10} \end{vmatrix} - (a_{11}-\Delta x_{m-1}a_{01})\begin{vmatrix} a_{00} & b_0-x_{m-1}a_{00} \\ a_{10} & b_1-x_{m-1}a_{10} \end{vmatrix}.$$

Il suffit alors d'expliciter b_0, b_1, a_{00}, a_{01}, a_{10} et a_{11} pour obtenir:

$$M_m^{(0)} = (x_m-x_0)\begin{vmatrix} \Sigma(x_j-x_{m-1}) & \Sigma(\Delta x_j-\Delta x_{m-1}) \\ \Sigma\Delta x_j(x_j-x_{m-1}) & \Sigma\Delta x_j(\Delta x_j-\Delta x_{m-1}) \end{vmatrix} + \Sigma\Delta x_j(\Delta x_j-\Delta x_{m-1})\begin{vmatrix} m & \Sigma(x_j-x_{m-1}) \\ \sum\Delta x_j & \Sigma(x_j-x_{m-1})\Delta x_j \end{vmatrix}$$

et, après des manipulations mineures:

$$M_m^{(0)} = (x_m-x_0)\begin{vmatrix} \Sigma(x_{m-1}-x_j) & \Sigma(x_m-x_{j+1}) \\ \Sigma\Delta x_j(x_{m-1}-x_j) & \Sigma\Delta x_j(x_m-x_{j+1}) \end{vmatrix} + \Sigma\Delta x_j(\Delta x_{m-1}-\Delta x_j)\begin{vmatrix} m & \Sigma x_j \\ \sum\Delta x_j & \Sigma x_j\Delta x_j \end{vmatrix}.$$

Les expressions $M_m^{(n)}$ se déduisent aisément des expressions $M_m^{(0)}$ en ajoutant n à tous les indices.

5.4 Cas des suites géométriques.

5.4.1 Proposition 8. *Si (x_n) est une suite géométrique de raison λ, alors:*

$$\frac{\partial}{\partial x_{n+p}} C_l^{(m)}(x_n) = \frac{\mu_p}{\delta^2 \Lambda}, \quad p=0,\dots,m,$$

avec:

$$\delta = (\lambda-1)(\lambda^m-1)$$

$$\mu_0 = \frac{\lambda^2(\lambda^m-1)^2(\lambda-1)(\lambda^{m-1}-1)}{\lambda+1}$$

$$\mu_m = \frac{(\lambda^m-1)^2(\lambda-1)(\lambda^{m-1}-1)}{\lambda+1}$$

$$\mu_l = (\lambda-1)^2(\lambda^m-1)^2\left(\frac{\lambda^m+1}{\lambda+1} - (\lambda+1)\lambda^{l-1}\right), \quad l=1,\dots,m-1.$$

$$\Lambda = m\frac{\lambda^m+1}{\lambda+1} - \frac{\lambda^m-1}{\lambda-1}.$$

Démonstration. Si nous posons $e_n = x_n - \hat{x}$, où $\hat{x}$ est le centre de la suite géométrique, nous avons: $e_{l+1} = \lambda e_l$, $\forall \lambda \in \mathbb{N}$. Un calcul simple mais fastidieux nous montre que pour cette suite, nous avons les 18 relations suivantes:

$$x_{n+m} - x_n = \Sigma\Delta x_l = e_n(\lambda^m-1),$$

$$x_{n+m} + x_n = 2\hat{x} + e_n(\lambda^m+1),$$

$$\Sigma x_l = m\hat{x} + \frac{\lambda^m-1}{\lambda-1}e_n$$

$$\Sigma x_{l+1} = m\hat{x} + \frac{\lambda^m-1}{\lambda-1}\lambda e_n$$

$$\Sigma(x_l+x_{l+1}) = 2m\hat{x} + \frac{(\lambda+1)(\lambda^m-1)}{\lambda-1}e_n$$

Annexe 1: transformation de suites par approximation.

$$\Sigma(x_l - x_n) = \left(\frac{(\lambda^m-1)}{\lambda-1} - m\right) e_n$$

$$\Sigma(x_{l+1} - x_{n+1}) = \left(\frac{(\lambda^m-1)}{\lambda-1} - m\right) \lambda e_n$$

$$\Sigma(x_{n+m-1} - x_l) = \left(m\lambda^{m-1} - \frac{\lambda^m-1}{\lambda-1}\right) e_n$$

$$\Sigma(x_{n+m} - x_{l+1}) = \left(m\lambda^{m-1} - \frac{\lambda^m-1}{\lambda-1}\right) \lambda e_n$$

$$\Sigma(\Delta x_l)^2 = (\lambda-1)(\lambda^m-1)\frac{\lambda^m+1}{\lambda+1}(e_n)^2$$

$$\Sigma(x_l \Delta x_l) = \hat{x}(\lambda^m-1)e_n + (\lambda^m-1)\frac{\lambda^m+1}{\lambda+1}(e_n)^2$$

$$\Sigma(x_{l+1} \Delta x_l) = \hat{x}(\lambda^m-1)e_n + (\lambda^m-1)\frac{\lambda^m+1}{\lambda+1}\lambda(e_n)^2$$

$$\Sigma\Delta x_l(x_l - x_n) = \lambda(\lambda^m-1)\frac{\lambda^{m-1}-1}{\lambda+1}(e_n)^2$$

$$\Sigma\Delta x_l(x_{l+1} - x_{n+1}) = \lambda^2(\lambda^m-1)\frac{\lambda^{m-1}-1}{\lambda+1}(e_n)^2$$

$$\Sigma\Delta x_l(x_{n+m-1} - x_l) = (\lambda^m-1)\frac{\lambda^{m-1}-1}{\lambda+1}(e_n)^2$$

$$\Sigma\Delta x_l(\Delta x_l - \Delta x_n) = \lambda(\lambda-1)(\lambda^m-1)\frac{\lambda^{m-1}-1}{\lambda+1}(e_n)^2$$

$$\Sigma\Delta x_l(\Delta x_{n+m-1} - \Delta x_l) = (\lambda-1)(\lambda^m-1)\frac{\lambda^{m-1}-1}{\lambda+1}(e_n)^2.$$

D'où les six déterminants:

$$D_1 = \begin{vmatrix} m & x_{n+m} - x_n \\ x_{n+m} - x_n & \Sigma(\Delta x_l)^2 \end{vmatrix} = \begin{vmatrix} m & e_n(\lambda^m-1) \\ e_n(\lambda^m-1) & (\lambda-1)(\lambda^m-1)\frac{\lambda^m+1}{\lambda+1}(e_n)^2 \end{vmatrix}$$

$$= (\lambda-1)(\lambda^m-1)(e_n)^2 \begin{vmatrix} m & \frac{\lambda^m-1}{\lambda-1} \\ 1 & \frac{\lambda^m+1}{\lambda+1} \end{vmatrix} = (e_n)^2 \delta\Lambda,$$

où: $\delta = (\lambda-1)(\lambda^m-1)$ et $\Lambda = m\frac{\lambda^m+1}{\lambda+1} - \frac{\lambda^m-1}{\lambda-1}$.

$$D_2 = \begin{vmatrix} 1 & x_n + x_{n+m} \\ m & \Sigma(x_l + x_{l+1}) \end{vmatrix} = \begin{vmatrix} 1 & 2\hat{x} + e_n(\lambda^m+1) \\ m & 2m\hat{x} + \frac{(\lambda+1)(\lambda^m-1)}{\lambda-1}e_n \end{vmatrix} = -e_n(1+\lambda)\Lambda.$$

$$D_3 = \begin{vmatrix} m & \Sigma x_l \\ \Sigma\Delta x_l & \Sigma x_l \Delta x_l \end{vmatrix} = \begin{vmatrix} m & m\hat{x} + \frac{\lambda^m-1}{\lambda-1}e_n \\ e_n(\lambda^m-1) & \hat{x}(\lambda^m-1)e_n + (\lambda^m-1)\frac{\lambda^m+1}{\lambda+1}(e_n)^2 \end{vmatrix} = (e_n)^2(\lambda^m-1)\Lambda.$$

$$D_4=\begin{vmatrix} m & \Sigma x_{i+1} \\ \Sigma\Delta x_i & \Sigma x_{i+1}\Delta x_i \end{vmatrix}=\begin{vmatrix} m & m\hat{x}+\frac{\lambda^m-1}{\lambda-1}\lambda e_n \\ e_n(\lambda^m-1) & \hat{x}(\lambda^m-1)e_n+(\lambda^m-1)\frac{\lambda^m+1}{\lambda+1}\lambda(e_n)^2 \end{vmatrix}=(e_n)^2(\lambda^m-1)\lambda\Lambda.$$

$$D_5=\begin{vmatrix} \Sigma(x_i-x_n) & \Sigma(x_{i+1}-x_{n+1}) \\ \Sigma\Delta x_i(x_i-x_n) & \Sigma(\Delta x_i)(x_{i+1}-x_{n+1}) \end{vmatrix}=\begin{vmatrix} (\frac{(\lambda^m-1)}{\lambda-1}-m)e_n & (\frac{(\lambda^m-1)}{\lambda-1}-m)\lambda e_n \\ \lambda(\lambda^m-1)\frac{\lambda^{m-1}-1}{\lambda+1}(e_n)^2 & \lambda^2(\lambda^m-1)\frac{\lambda^{m-1}-1}{\lambda+1}(e_n)^2 \end{vmatrix}=0.$$

$$D_6=\begin{vmatrix} \Sigma(x_{n+m-1}-x_i) & \Sigma(x_{n+m}-x_{i+1}) \\ \Sigma\Delta x_i(x_{n+m-1}-x_i) & \Sigma(\Delta x_i)(x_{n+m}-x_{i+1}) \end{vmatrix}=\begin{vmatrix} (m\ \lambda^{m-1}-\frac{\lambda^m-1}{\lambda-1})\ e_n & (\lambda^m-1)\frac{\lambda^{m-1}-1}{\lambda+1}(e_n)^2 \\ (m\ \lambda^{m-1}-\frac{\lambda^m-1}{\lambda-1})\lambda e_n & \lambda(\lambda^m-1)\frac{\lambda^{m-1}-1}{\lambda+1}(e_n)^2 \end{vmatrix}=0.$$

Compte tenu de ces six déterminants, la proposition 7 donne:

$$\frac{\partial}{\partial x_{n+p}}C_1^{(m)}(x_n)=\frac{M_p^{(n)}}{(D^{(n)})^2},\ p=0,\dots,m,$$

avec:

$$D^{(n)}=(e_n)^2\delta\Lambda$$

$$M_0^{(n)}=\lambda(\lambda-1)(\lambda^m-1)\frac{\lambda^{m-1}-1}{\lambda+1}(e_n)^2\times\lambda(\lambda^m-1)\Lambda(e_n)^2=(e_n)^4\mu_0\Lambda.$$

$$M_m^{(n)}=(\lambda-1)(\lambda^m-1)\frac{\lambda^{m-1}-1}{\lambda+1}(e_n)^2\times(\lambda^m-1)\Lambda(e_n)^2=(e_n)^4\mu_m\Lambda.$$

$$M_p^{(n)}=(e_n)^2\delta(\lambda-1)(\lambda^m-1)\frac{\lambda^m+1}{\lambda+1}(e_n)^2-(e_n)^2(\lambda^m-1)^2\times(\lambda-1)^2\lambda^{p-1}e_n\times e_n(1+\lambda)\Lambda$$

$$=(e_n)^4\mu_p\Lambda,\ \text{pour } p=1,\dots,m-1.$$

Après simplification par $(e_n)^2\Lambda$, on obtient: $\frac{M_p^{(n)}}{(D^{(n)})^2}=\frac{\mu_p}{\delta^2\Lambda}$, pour $p=0,\dots,m$.

5.4.2 Nombre de conditionnement.

Corollaire: *Le nombre de conditionnement ponctuel du procédé $C_1^{(m)}$ vis à vis d'une suite géométrique (x_i) de raison λ est égal à:* $\Gamma(C_1^{(m)},(x_i),n)=\dfrac{\left|\frac{1+\lambda^2}{1-\lambda^2}(\lambda^{m-1}-1)\right|+\sum_{i=1}^{m-1}\left|\frac{\lambda^m+1}{\lambda+1}-(\lambda+1)\lambda^{i-1}\right|}{m\frac{\lambda^m+1}{\lambda+1}-\frac{\lambda^m-1}{\lambda-1}}$

Annexe 1: transformation de suites par approximation.

Démonstration: Par définition du nombre de conditionnement ponctuel:

$$\Gamma(C_1^{(m)},(x_i),n) = \sum_{p=0}^{m} \left| \frac{\partial}{\partial x_{n+p}} (C_k^{(m)}(x_n)) \right|.$$

D'après la proposition 8: $$\sum_{p=0}^{m} \left| \frac{\partial}{\partial x_{n+p}} (C_k^{(m)}(x_n)) \right| = \frac{|\mu_0| + \sum_{p=1}^{m-1} |\mu_p| + |\mu_m|}{\delta^2 |\Lambda|}.$$

Il suffit de noter que l'on a les trois relations suivantes pour aboutir au résultat annoncé:

$$|\Lambda| = \Lambda,$$

$$|\mu_0| + |\mu_m| = (\lambda^m - 1)^2 \frac{(\lambda^{m-1} - 1)(\lambda - 1)}{|\lambda + 1|} (1 + \lambda^2)$$

$$\sum_{p=1}^{m-1} |\mu_p| = (1-1)^2 (\lambda^m - 1)^2 \sum_{p=1}^{m-1} \left| \frac{\lambda^m + 1}{\lambda + 1} - (\lambda + 1)\lambda^{p-1} \right|$$

Remarque. Puisque ce nombre de conditionnement ponctuel est indépendant de l'indice n, il s'ensuit que le nombre de conditionnement global est défini et lui est égal.

5.5 Suites à convergence géométrique.

5.5.1 Nombre de conditionnement.

Proposition 9. *Le nombre de conditionnement global de la transformation $C_k^{(m)}$ vis à vis de la suite (x_n) à convergence géométrique de raison λ est égal à son nombre de conditionnement global vis à vis de toute suite géométrique de même raison.*

Démonstration. D'après la proposition 7, nous avons: $\frac{\partial}{\partial x_{n+p}} C_1^{(m)}(x_n) = \frac{M_p^{(n)}}{(D^{(n)})^2}$, p=0,...,m.

Puisque la suite est à convergence géométrique, il existe une suite (ε_n) de limite nulle telle que: $x_{n+1} - \hat{x} = (1+\varepsilon_n)(x_n - \hat{x})$, où $\hat{x}$ est la limite de la suite (x_n).

Si nous utilisons les nombres δ et $\mu_1, \ldots, \mu_m$ ($l=0,\ldots,m$) introduits dans la proposition 8, nous avons: $\lim_{n\to\infty} \frac{D^{(n)}}{(e_n)^2} = \delta\Lambda$ et $\lim_{n\to\infty} \frac{M_l^{(n)}}{(e_n)^4} = \mu_l \Lambda$, pour $l=0,\ldots,m$.

En effet, chacun des nombres $\frac{D^{(n)}}{(e_n)^2}$ et $\frac{M_l^{(n)}}{(e_n)^4}$ s'écrit sous la forme d'un polynôme en fonction des m+1 variables ε_j ($j=n,\ldots,n+m-1$), et ces polynômes ont précisément les valeurs $\delta\Lambda$ et $\mu_l\Lambda$ aux points ε_j, $j=n,\ldots,n+m-1$. La continuité de la fonction polynôme assure donc le résultat.

Il s'ensuit que le nombre de conditionnement $\Gamma(C_1^{(m)},(x_i),n)=\sum_{l=0}^{m}\frac{|M_l^{(n)}|}{(D^{(n)})^2}$ tend vers $\sum_{l=0}^{m}\frac{|\mu_l|}{\delta^2\Lambda}$,

c'est à dire vers le nombre de conditionnement global de toute suite géométrique de même raison.

5.5.2 Limite du nombre de conditionnement.

Proposition 10 *Lorsque m tend vers l'infini, le nombre de conditionnement global du procédé $C_1^{(m)}$ vis à vis d'une suite à convergence géométrique tend vers 1.*

Démonstration: D'après la proposition 9, ce nombre de conditionnement est égal à:

$$\Gamma(C_1^{(m)},(x_i),n)=\frac{\left|\frac{1+\lambda^2}{1-\lambda^2}(\lambda^{m-1}-1)\right|+\sum_{l=1}^{m-1}\left|\frac{\lambda^m+1}{\lambda+1}-(\lambda+1)\lambda^{l-1}\right|}{m\,\frac{\lambda^m+1}{\lambda+1}-\frac{\lambda^m-1}{\lambda-1}}.$$

Le numérateur est majoré par le nombre

$$N_m=\frac{1+\lambda^2}{1-\lambda^2}(\lambda^{m-1}-1)+(1+\lambda)\frac{1-\lambda^{m-1}}{1-\lambda}+(m-1)\frac{1+\lambda^m}{1+\lambda}=(m-1)\frac{1+\lambda^m}{1+\lambda}+2\,\frac{1+\lambda+\lambda^2}{1+\lambda}\times\frac{1-\lambda^{m-1}}{1-\lambda}.$$

Soit m_0 le plus petit entier tel que: $|\lambda^{m-1}|<\frac{1}{2}$. Alors les deux rapports $\frac{1-\lambda^m}{1+\lambda^m}$ et $\frac{1-\lambda^{m-1}}{1+\lambda^m}$ sont compris entre $\frac{1}{3}$ et 3. Le nombre ε étant fixé, soit m_1 le plus petit entier supérieur ou égal à $\frac{6(1+\lambda+\lambda^2)}{1-\lambda}\times\frac{1}{\varepsilon}$. Pourvu que $m\geq\max(m_0,m_1)$, nous avons:

$$N_m=(m-1)\frac{1+\lambda^m}{1-\lambda}\times\left(1+2\,\frac{1+\lambda+\lambda^2}{1-\lambda}\times\frac{1-\lambda^{m-1}}{1+\lambda^m}\times\frac{1}{m-1}\right).$$

D'où les majorations successives:

$$N_m<(m-1)\frac{1+\lambda^m}{1-\lambda}\times\left(1+6\,\frac{1+\lambda+\lambda^2}{1-\lambda}\times\frac{1}{m-1}\right)\quad\text{car } m\geq m_0,$$

$$N_m<(m-1)\frac{1+\lambda^m}{1-\lambda}\times(1+\varepsilon)\quad\text{car } m\geq m_1.$$

De même le dénominateur $D_m=m\frac{\lambda^m+1}{\lambda+1}-\frac{\lambda^m-1}{\lambda-1}=m\frac{\lambda^m+1}{\lambda+1}\left(1-\frac{1+\lambda}{1-\lambda}\times\frac{1}{m}\times\frac{1-\lambda^m}{1-\lambda^m}\right)$ est minoré par

$m\frac{\lambda^m+1}{\lambda+1}\left(1-\frac{1+\lambda}{1-\lambda}\times\frac{3}{m}\right)$ (car $m\geq m_0$), lui-même minoré par $m\frac{\lambda^m+1}{\lambda+1}\left(1-\frac{\varepsilon}{2}\right)$ (car $m\geq m_1$).

Il s'ensuit que, pour tout ε positif, il existe un entier $m_2=\min(m_0,m_1)$ tel que $m\geq m_2$ assure:

$$\Gamma(C_1^{(m)},(x_i),n)<\frac{1+\varepsilon}{1-\varepsilon/2}\left(1-\frac{1}{m}\right),\ \text{d'où:}\ \overline{\lim_{n\to\infty}}\,\Gamma(C_1^{(m)},(x_i),n)\leq 1.$$

D'autre part, la translativité du procédé implique: $\overline{\lim_{n\to\infty}}\,\Gamma(C_1^{(m)},(x_i),n)\geq 1$.

6. Résultats numériques.

Pour illustrer cette dernière propriété, nous appliquons le procédé $C_1^{(m)}$ à des suites $(x_n)_{n\geq 0}$ de la forme: $x_{n+1}-\hat{x} = \lambda\,(x_n-\hat{x})(1+\varepsilon_n)$, où $|\lambda| \in]0,1[$, et où $(\varepsilon_n)_{n\geq 0}$ est une suite de nombres aléatoires dont la loi de répartition est uniforme sur l'intervalle $[-\varepsilon,\varepsilon]$. La suite $(x_n)_{n\geq 0}$ est donc une suite à convergence géométrique de raison λ et de limite $\hat{x}$.

Pour λ fixé, nous tabulerons en fonction de m les quantités $M_m^{(n)}(\varepsilon) = \dfrac{\max\limits_{i=0\ldots m-i} \left| C_1^{(m)}(x_i)-\hat{x} \right|}{\varepsilon}$ obtenues avec des suites (x_n) correspondant à diverses valeurs de n et de ε. Ces valeurs seront placées en regard du nombre de conditionnement $\Gamma(C_1^{(m)},(x_i),n)$ dépendant de m, de λ et de n. Les résultats sont présentés dans les deux tableaux 1 et 2.

L'analyse des résultats montre que les nombres $M_m^{(n)}(\varepsilon)$ sont effectivement majorés par le nombre de conditionnement auquel ils se rapportent, et que la majoration n'est jamais plus de dix fois supérieure à la valeur effective du quotient $M_m^{(n)}(\varepsilon)$: le nombre de conditionnement retenu est un bon critère de mesure de la sensibilité de la suite transformée aux perturbations de la suite initiale.

Dans le tableau n°3, on trouvera, multipliées par 10^8, les valeurs de l'écart $C_1^{(m)}(x_n) - \hat{x}$ pour m=2, ... ,9. La propriété de lissage de la transformation apparaît alors clairement: l'incidence de la perturbation est 10 fois plus faible pour m=8 ou 9 que pour m=2.

Tableau 1: λ=0,9

m	$\Gamma(C_1\ (x_n))$	$\varepsilon=10^{-4}/2$		$\varepsilon=10^{-6}/2$	
		suite n°1 (n=12)	suite n°2 (n=20)	suite n°1 (n=20)	suite n°2 (n=40)
2	361	125	59,2	78,3	83,1
3	180	59,4	37,1	47,9	46,4
4	108	40,1	25,6	31,3	20,3
5	71,5	21,0	18,4	25	13,6
6	50,8	14,7	15,5	13,8	5
7	38	10,7	14,2	9,71	4,62
8	29,3	10,1	8,91	9,28	4,42
9	23,3	5,22	4,91	5,72	4,34
10	18,9	4,28	3,93	5,78	3,38
11	15,7	1,62	3,35	3,87	2,77
12	13,17	1,44	3,20	2,34	2,66

Tableau 2: λ=0,99

m	$\Gamma(C_1\ (x_n))$	$\varepsilon=10^{-4}/2$		$\varepsilon=10^{-6}/2$		$\varepsilon=10^{-12}/2$
		suite n°1 (n=12)	suite n°2 (n=20)	suite n°1 (n=20)	suite n°2 (n=40)	suite n°3 (n=20)
2	39601	12758	11771	17319	15899	14379
3	19800	6596	7712	7132	7888	8182
4	11879	3566	5790	3976	6163	4421
5	7919	2236	4422	2866	4265	3147
6	5656	2333	2789	1641	3163	2542
7	4242	1793	2452	1632	2598	2350
8	3299	1631	2281	1338	1422	1183
9	2639	1352	1621	574	1354	1286
10	2159	1103	1300	692	1261	1301
11	1799	832	752	699	1253	799
12	1522	808	692	721	834	658

Tableau 3: λ=0.99, $\varepsilon=10^{-8}/2$

n	x_n	$C_{1,2}(x_n)$	$C_{1,3}(x_n)$	$C_{1,4}(x_n)$	$C_{1,5}(x_n)$	$C_{1,6}(x_n)$	$C_{1,7}(x_n)$	$C_{1,8}(x_n)$	$C_{1,9}(x_n)$
00	1.0000	3748.0	942.2	1549.0	- 229.0	- 258.9	269.2	532.9	567.1
01	0.9900	-1864.0	1018.0	-1384.0	- 963.4	- 24.5	398.5	474.9	298.3
02	0.9801	3899.0	-2025.0	-1074.0	285.0	745.8	744.8	456.6	286.1
03	0.9703	-7950.0	-2414.0	294.0	935.5	870.9	464.3	249.3	- 80.5
04	0.9606	3121.0	3944.0	3081.0	2133.0	1170.0	692.0	169.1	- 131.6
05	0.9510	4768.0	2781.0	1508.0	410.2	26.0	- 456.0	- 675.6	- 283.4
06	0.9415	793.9	- 2.8	- 894.2	- 910.3	-1241.0	-1299.0	- 652.9	- 677.2
07	0.9321	- 799.6	-1754.0	-1348.0	-1636.0	-1581.0	- 670.2	- 696.4	- 578.0
08	0.9227	-2709.0	-1396.0	-1804.0	-1661.0	- 419.0	- 524.4	- 419.5	- 518.7
09	0.9135	- 83.1	-1638.0	-1506.0	170.3	- 166.6	- 134.0	- 329.4	39.1
10	0.9044	-3193.0	-1937.0	0793.2	72.4	36.3	- 263.6	180.5	122.3
11	0.8953	- 680.6	3032.0	831.3	477.2	- 69.6	446.0	306.9	309.1
12	0.8864	6743.0	601.8	185.4	- 463.3	355.3	202.3	231.2	
13	0.8775	-5541.0	-2140.0	-1950.0	- 209.4	- 225.1	- 65.8		
14	0.8687	1261.0	- 690.2	1291.0	668.8	0583.2			
15	0.8601	-2642.0	1961.0	693.0	563.5				
16	0.8515	6564.0	1382.0	0843.2					
17	0.8429	-3801.0	1243.0						
18	0.8345	1314.0							
19	0.8262								
20	0.8179								
valeur maxi.		7950.0	3944.0	3081.0	2133.0	1581.0	1299.0	696.4	677.2

7. Conclusion

La substitution d'une technique d'approximation à la technique d'interpolation nous a permis de proposer une nouvelle classe de transformations non linéaires de suites de nombres réels ou complexes. Une étude plus précise de la plus simple de ces transformations, un procédé qui généralise très naturellement le procédé δ^2 d'Aitken montre que ce nouveau procédé C_1^m conserve les propriétés du procédé δ^2 reposant sur l'interpolation qu'il généralise (quasi-régularité et accélération des suites à convergence géométrique). En outre, ce procédé est d'autant mieux conditionné qu'il fait appel à davantage de termes de la suite initiale et son nombre de conditionnement global relatif à une suite à convergence géométrique tend vers l'unité quand le nombre de termes utilisés augmente indéfiniment. En d'autres termes, cette transformation fournit des estimations de la limite de la suite initiale dans lesquelles l'incertitude qui affecte les termes de la suite initiale est d'autant moins amplifiée que le nombre des termes utilisés pour obtenir ces estimations est élevé. Les expériences numériques auxquelles nous avons procédé viennent confirmer ce résultat. Cette méthode semble donc bien adaptée à l'objectif poursuivi. Toutefois, on étude a été limitée ici au cas où m= 1. Il convient maintenant de vérifier les résultats prometteurs auxquels nous avons abouti s'étendent à l'extension de la transformation de Shanks.

8. Références

[1] **Brezinski C.**-Etudes sur les ε et ρ-algorithmes, Numer. Math. 17 (1971) 153-162.

[2] **Brezinski C.**-Accélération de la convergence en analyse numérique, Lect. Notes in Math.584, Springer-Verlag (1977).

[3] **Cordellier F.**-Sur la régularité des procédés δ^2 d'Aitken et W de Lubkin. Padé Approximation and its applications, L. Wuytack ed., L. N. M. 765, Springer Verlag, Berlin (1979) 20-35.

[4] **Gantmacher F.R.**-The theory of matrices, Vol 1, Chelsea New-York (1960).

[5] **Germain-Bonne B.**-Transformations de suites, RAIRO (1973) 84-90.

[6] **Gray H.L.and Clark W.D.**-On a class of nonlinear transformations and their applications to the evaluation of infinite series, Journ. of Res. of the N.B.S. 73B (1969) 251-274.

[7] **Lubkin S.**-A Method of Summing Infinite Series, Journ.of Res. of the N.B.S. 48 (1952) 228-254.

[8] **Marx I.**- Remark concerning a nonlinear sequence-to-sequence transform. J. Math. Phys. 42 (1963) 334-335.

[9] **Rice J.R.**- Sequence transformations based on Tchebycheff approximations. J. Res. N.B.S. 64B (1960) 227-235.

[10] **Rice J.R.**- A theory of condition. S.I.A.M. Journ. Numer. Anal. 3 (1966) 287-310.

[11] **Shanks D.**-Non linear transformations of divergent and slowly convergent series, J. Math. Phys. 34 (1955) 1-42.

[12] **Streit R.F.**- The T_{+m} transformation. Math. Comp. 30 (1976) 505-511.

[13] **Tucker R.R.**- The δ^2-process and related Topics, Pacific J. Math. 22 (1967) 349-359 (part 1) and 28 (1969) 455-463 (part 2).

[14] **Wynn P.**- On a device for computing the $e_m(s_n)$ transformation, M.T.A.C. 10 (1956) 91-96.

[15] **Wynn P.**- A sufficient condition for the instability of the ε-algorithm, Nieuw. Arch. Wisk. 3 (1961) 117-119.

[16] **Wynn P.**- On the convergence and stability of the ε-algorithm, SIAM J. Numer. Anal. 3 (1966) 91-122.

On the use of Kronecker's algorithm in the generalized rational interpolation problem.

Florent Cordellier

Abstract

Kronecker's algorithm can be used to solve the generalized rational interpolation problem. In order to present the algorithm, rational forms are used in stade of too restrictive rational fractions. The proposed algorithm is totally reliable as soon as the functionals that characterize the problem satisfy two precise conditions. These conditions are fulfilled in the modified Hermite rational interpolation problem and, as a consequence, in the special case of the Cauchy problem and in the Padé approximant problem. This reliability covers two properties : on one hand, every rational form given by the algorithm is a solution of the problem whereas, on the other hand, every solution of the problem is given by the algorithm (with the exception of an eventual reduction of the rational form). However, if the algorithm gives a non reduced rational form, then the corresponding rational fraction is not a solution of the problem.

0. Introduction

Introduced by Kronecker [9] as soon as 1881, the algorithm discussed here is certainly the most elegant among whole rational interpolation algorithms [12]. Some papers have recently shown that the feasible scope of this algorithm can be widely extended without giving precise limits to such an extension : for instance Meinguet [11] has shown that this algorithm still gives good solution even when there are unattainable points whereas Warner [12] says without proving that this algorithm can be applied to the modified rational Hermite problem. At last, Graves-Morris [8] gives a description of a reliable Kronecker's algorithm for solving Padé and Cauchy problem but without taking account of the modified Hermite problem. In the special case of Padé problem, Baker has recalled this algorithm and given an interessing modified form known as Baker's algorithm [1], but it has been necessary to wait for 1978 for applying these algorithms to non normal Padé table : then a reliable division algorithm has been proposed independently by Mc Eliece and Shaerer [10] and Cordellier [4] ; this last unpublished paper also contains a reliable form of Baker's algorithm. Claessens gives interesting results dealing with the Rational Hermite Table [3].

The aim of this paper is to present the division algorithm in such a manner that it will be reliable on a wide class of rational interpolation problems, and to precise the exact meaning of this reliability.

As the outset some basic concepts are recalled : in order to distinguish (reduced) rational fractions from (reducible) rational form, Gilewicz's terminology is used. As it will be seen, division algorithm deals with rational forms in place of rational fractions. The rational interpolation problem is proposed in a enough general way in order to cover modified Hermite rational interpolation problem [12] by using linear functionals similar to those used by Davis [6] in polynomial interpolation problems. Paragraph 2 is used to present a foundamental operation in the description of division algorithm : this operation is refered to as quotient of rational forms. In the following paragraph are given the definition and some properties of the inverse of a rational form and of derived notions. Paragraph 4 and 5 are respectively used to present two different forms of the division algorithm. The direct form is the one that is actually used and it is shown that, provided that a correct initialization is known, this algorithm gives rational forms satisfying the initial problem. Inverse form is a theoretical algorithm used to clear up properties of the direct form : thanks to its use, it can be shown that there is no other actual solution of the problem that the ones given by direct algorithm.

1. Position of the problem

1.1 - Polynomials, rational forms and fractions.

Let us recall some notions used in the paper. These notions are classical enough for allowing us to be short, and the reader is refered Gilewicz [7] for more details.

Here K is a commutative field, the characteristic of which is infinite, $\mathbf{R}$ or $\mathbf{C}$ for instance. For every $n \in \mathbf{N}$, let us denote by I_n, I_n^* and I_n' the following finite

sets :

$$I_n = \{i \in N | i \leq n\}, \quad I_n^* = I_n \backslash \{0\}, \quad I_n' = I_n \cup \{-1\}.$$

Let us denote by $\mathcal{P}_n$ the linear space of polynomials with coefficients in K and degree not greater than n, with $\mathcal{P}_{-1} = \{0\}$. This allows us to denote, for every $n \in \mathbf{N}$: $\mathcal{P}_n^* = \mathcal{P}_n \backslash \mathcal{P}_{n-1}$. This notation is completed by $\mathcal{P}_{-1}^* = \mathcal{P}_{-1}$. Moreover $\mathcal{P}$ is the linear space of all polynomials : $\mathcal{P} = \cup_{i=-1}^{\infty} \mathcal{P}_i = \cup_{i=-1}^{\infty} \mathcal{P}_i^*$. Let us set the degree of a null polynomial to be equal to -1 (and not $-\infty$ as usual). In order to simplity the paper, let us use the following convention : polynomials will be represented by capital letters P, Q, A and B (possibly identified by some index or mark) whereas their degree will be represented by the corresponding small letter (identified by the same index or mark).

Let us denote by $\mathcal{B}$ the product space $\mathcal{P} \times \mathcal{P}$: each element of $\mathcal{B}$ is an ordered pair of polynomials. The degree of such a pair (P, Q) will be denoted by (p, q).

Let us define on $\mathcal{B}$ an equivalence relation $\sim$ by :

$$(P, Q) \sim (P', Q') \iff \exists k \in K \backslash \{0\} \text{ such that } P' = AP \text{ and } Q' = AQ.$$

This relation induces on $\mathcal{B}$ a quotient space : it is the space of rational forms. So a rational form can be seen as a pair of polynomials with that exception that the product of both polynomials by some non null constant doesn't affect this form. Then a rational form will be represented by two polynomials the first of them is the numerator, the second one is the denominator. It is possible to be more precise by introducing a normalization in order to have a unic representation of rational forms by a suitably normed pair of polynomials, but this normalization makes our presentation too heavy and it has been droped here.

Let us introduce a second equivalence relation $\approx$ on $\mathcal{B}$ by :

$$(P, Q) \approx (P', Q') \iff PQ' = P'Q$$

Because it is compatible with the first one, this new relation induces on the set of rational forms a quotient space : it is the space of rational fractions. A rational fraction can be seen as a rational form the numerator and the denominator of which are relatively prime. Such a rational form is reduced.

1.2 - Functionals.

Let us denote by :

X : a non empty part of E, a commutative algebra on K.

$\mathcal{E}$: the linear space consisting of functions from X into E.

A commutative inner product can be defined in $\mathcal{E}$ by :

$$h = f \times g \iff x \in X \rightarrow h(x) = f(x) \times g(x)$$

so that $\mathcal{E}$ is a commutative algebra on K, containing $\mathcal{P}$ as a subspace.

$\mathcal{F}$: a subspace of $\mathcal{E}$ containing $\mathcal{P}$.

$\mathcal{F}^*$: its algebraic dual, consisting of the set of linear functionals on K.

Σ_n : a finite dimensional subspace of $\mathcal{F}^*$, with dimension $n+1$.

β_i, $i \in I_n$: a basis of Σ_n.

$\mathcal{N}(\Sigma_n) = \{f \in \mathcal{F} | \Sigma \in \Sigma_n \rightarrow \Sigma(f) = 0\}$, the kernel of Σ_n.

$\overset{(n)}{C}$: the $(n+1) \times (n+1)$ square matrix, the general term in row $i \in I_n$ and in column $j \in I_n$ is given by : $\overset{(n)}{C}{}_i^j = \beta_i(x^j)$

Let us introduce now the two following assumptions :

$(\mathcal{R})$ $\overset{(n)}{C}$ is regular (regularity condition)

$(\mathcal{S})$ $P \in \mathcal{P}$ and $f \in \mathcal{N}(\Sigma_n) \Rightarrow Pf \in \mathcal{N}(\Sigma_n)$ (stability condition)

Remark 1. If every polynomial $P \in \mathcal{P}$ can be written as a product of polynomials the degree of which is no greater than d, then in hypothesis $(\mathcal{S})$ the condition $P \in \mathcal{P}$ can be replaced by $P \in \mathcal{P}_d$. For example, for $K = \mathbf{R}$, one has $d = 2$ whereas for $K = \mathbf{C}$, one has $d = 1$.

1.3 - The general rational interpolation problem.

Definition. *Let Σ_n a (n+1)-dimensional subspace of $\mathcal{F}^*$ (the dual of $\mathcal{F}$), satisfying both conditions $(\mathcal{R})$ and $(\mathcal{S})$. A rational form $R = (P, Q)$ is said to be an interpolant of $f \in \mathcal{F}$ with respect to Σ_n if and only if :*

$$\Sigma \in \Sigma_n \Rightarrow \Sigma(P - fQ) = 0.$$

Problem. Under above conditions, find all rational forms $R = (P, Q)$ with degree (p, q) satisfying the two following conditions :

R *is an interpolant of f with respect to Σ_n* ;

$p + q \leq n$.

1.4 - The modified Hermite rational interpolation problem.

The above problem is a formal one. It is interesting to verify that it contains as a special case the important class of modified Hermite rational interpolation problems [12].

Let a part X of $K = \mathbf{R}$ or $\mathbf{C}$, and m couples $c_i = (x_i, e_i)$ with $x_i \in X$ and $e_i \in I_k$ satisfying : $e_i > 0 \Rightarrow x_i \in \overset{0}{X}$ for $i \in I_m^*$, where $\overset{0}{X}$ denotes the interior of X.

Let us choose $\mathcal{F}$ as the space of functions such that $f^{(j)}(x_i)$ is defined for $j \in I_{e_i}, i \in I_m^*$ (where $f^{(j)}$ represents the j^{th} derivative of f).

It is possible to define the functionals $\gamma_{i,j}$ by : $\gamma_{i,j}(f) = f^{(j)}(x_i), j \in I_{e_i}, i \in I_m^*$.

Proving that the functionals $\gamma_{i,j}$ are linearly independant is a classical exercise [6], so that the space Σ_n span by these functionals satisfies hypothesis $(\mathcal{R})$. By using Leibnitz formula, a straightforward calculus shows us that hypothesis $(\mathcal{S})$ is satisfied :

$$(f \in \mathcal{F}, i \in I_m^*, j \in I_{e_i} \Rightarrow \gamma_{i,j}(f) = 0 \text{ and } Q \in \mathcal{P}) \\ \Rightarrow (i \in I_m^*, j \in I_{e_i} \Rightarrow \gamma_{i,j}(Qf) = 0).$$

The classical Hermite rational interpolation problem consists of determining $R = (P, Q)$ with degree (p, q) satisfying :

$$i \in I_m^*, j \in I_{e_i} \Rightarrow \gamma_{i,j}(f - R) = 0$$
$$p + q + 1 \leq \Sigma_{i=1}^{m}(e_i + 1)$$

whereas in the modified Hermite rational interpolation problem, the condition $\gamma_{i,j}(f - R) = 0$ is replaced by $\gamma_{i,j}(P - fQ) = 0$, so that the modified Hermite problem is a special case of the general problem posed in 1.3.

As well known, the classical problem not always has a solution whereas the modified problem has. However, if the classical problem has a solution, then this solution is the same as the one of the modified problem [12]. Let us still recall [3] that modified the Hermite problem covers two classical problems : the Cauchy interpolation problem [11] is obtained with $i \in I_m^*$ and $e_i = 0$ for every $i \in I_m^*$ and the Padé approximant problem [1] corresponds to $m = 1$ and $x_1 = 0$.

2. Quotient of rational forms.

2.1 - Definitions.

Let two rational forms $R = (P, Q)$ and $R' = (P', Q')$ with respective degrees (p, q) and (p', q') satisfying $p' \geq 0$. When dividing P by P', let us denote the quotient by A and the rest by P'', with respective degrees a and p''. Then one has : $P = AP' + P''$ with $a = max(-1, p - p')$ and $p'' < p'$.

Let us denote by Q'' the polynomial with degree q'' defined by : $Q'' = Q - AQ'$.

The rational form $R'' = (P'', Q'')$ is the result of a binary inner operation on rational forms. This operation is called quotient and is denoted by the symbol % : $R'' = R \,\%\, R$.

2.2 - Elementary properties of quotient.

Proposition 1. *Let $R = (P, Q)$ and $R' = (P', Q')$ be two rational forms with respective degrees (p, q) and (p', q') with $p' \geq 0$, and $R'' = R \,\%\, R'$ their quotient with degree (p'', q'').*

(i) *besides $p'' < p'$, one has : $q'' \leq q' + \max(q - q', p - p')$.*

(ii) *moreover, if $q - q' \neq p - p'$, one has : $q'' - q = \max(q - q', p - p')$.*

(iii) $p > p' \wedge q \leq q' \Rightarrow q'' - q' = p - p' > 0$

(iv) $R = R \,\%\, R' \iff p < p'$

(v) $R \,\%\, R' = (0,0) \iff \exists A \neq 0$ *such that* $P = AP'$ *and* $Q = AQ'$.

Proof.

(i) From definition of P'', one has $P'' = P - AP'$ with $p'' < p'$. If $p' \leq p$ then $p = a + p'$, and from $Q'' = Q - AQ'$, one has $q'' \leq \max(q, q' + a) = \max(q, q' + p - p')$, else $A = 0$, so that $Q'' = Q$ and $q'' = q \leq \max(q, q' + p - p')$. Point (i) is true in both cases.

(ii) If $q - q' \neq p - p'$, then the respective degrees of Q and AQ' are distinct, and this insures $q'' = \max(q, q' + p - p')$.

(iii) From $p > p'$ and $q \leq q'$, one has : $q - q' \leq 0 < p - p'$; then from (ii) : $q'' = q' + p - p' > q'$.

(iv) $R = R \,\%\, R' \iff P - AP' = P \wedge Q - AQ' = Q \wedge p < p'$ (definition). From which the necessary and sufficient condition : $p < p'$.

(v) $R \,\%\, R' = (0,0) \iff P - AP' = 0 \wedge Q - AQ' = 0 \wedge -1 < a$ (definition). This means there is some polynomial $A \neq 0$ satisfying $P = AP'$ and $Q = AQ'$.

Remark 2. The quotient of two reduced rational forms is not necessarily reduced as it can be seen in the following example :

Let $R = ((x-3)^3, 1)$, $R' = ((x-3)(x-4), x-2)$ and $R'' = (x-3, -(x-3)(x+1))$. R and R' are reduced forms (and then can be interpreted as rational fractions) whereas $R'' = R \,\%\, R'$ is not reduced (and cannot be consider as a rational fraction without reducing).

This remark is essential when using the division algorithm because it allows us to see that this algorithm deals with rational forms and not with rational fractions.

2.3 - Foundamental property.

Proposition 2. *Let a rational interpolation problem with functionals* Σ_n *satisfying assumptions* $(\mathcal{R})$ *and* (S) *on* $\mathcal{F}$ *and a function* $f \in \mathcal{F} \setminus \mathcal{N}(\Sigma_n)$.

Let $R = (P, Q)$ *and* $R' = (P', Q')$ *be two rational forms with respective degrees* (p, q) *and* (p', q') *satisfying* $p' \neq -1$ *and* $R'' = (P'', Q'')$ *their quotient :* $R'' = R \,\%\, R'$. *If* R *and* R' *are two distinct interpolants of* f *with respect to* Σ_n, *then* R'' *is also an interpolant of* f *with respect to* Σ_n.

Proof. By definition of R'', one has $P'' = P - AP'$ and $Q'' = Q - AQ'$. Then $\Sigma \in \Sigma_n$ implies (thanks to the linearly of Σ) :

$$\Sigma(P'' - fQ'') = \Sigma((P - AP') - f(Q - AQ')) = \Sigma(P - fQ) - \Sigma(AP' - fAQ')$$

Our hypothesis insures $\Sigma(P - fQ) = \Sigma(P' - fQ') = 0$. The commutativity of K insures : $\Sigma(AP' - fAQ') = \Sigma(A(P' - fQ'))$. By assumption (S) :

$$\Sigma(P' - fQ') = 0 \Rightarrow \Sigma(A(P' - fQ')) = 0$$

and the previous result follows. ∎

3. Inverse of a rational form.

3.1 - Definitions.

The rational form obtained by exchanging numerator and denominator in a rational form $R = (P, Q)$ is called inverse of R and denoted by $R^{-1} = (Q, P)$.

Let $R = (P, Q)$ and $R' = (P', Q')$ be two rational forms with respective degrees (p, q) and (p', q') satisfying $q' \geq 0$. The inverse quotient R''' of the two forms R and R' is by definition the inverse of the quotient of their respective inverses R^{-1} and R'^{-1} : $R'''^{-1} = R^{-1} \,\%\, R'^{-1}$.

3.2 - Proporties of the inverse quotient.

Bearing in mind the definition of the inverse quotient, elementary and foundamental properties of quotient induce on inverse quotient corresponding properties. These properties are given here without their straightforward proofs.

Proposition 3. *Let $R''' = (P''', Q''')$ the inverse quotient of the two rational forms $R = (P, Q)$ by $R' = (P', Q')$ with respective degrees (p, q) and (p', q') satisfying $q' \geq 0$.*

(i) : besides $q''' < q'$, one has $p''' \leq p' + max(p - p', q - q')$

(ii) : moreover, if $q - q' \neq p - p'$, then $p''' - p' = max(p - p', q - q')$

(iii) : $q > q' \wedge p \leq p' \Rightarrow p'' - p' = q - q' > 0$.

(iv) : $R^{-1} = R^{-1} \,\%\, R'^{-1} \iff q < q'$

(v) : $R^{-1} \,\%\, R'^{-1} = (0, 0) \iff \exists A \neq 0$ such that $P = AQ'$ and $Q = AQ'$.

Proposition 4. *Let a rational interpolation problem with functional Σ_n satisfying assumptions $(\mathcal{R})$ and (S) on $\mathcal{F}$ and a function $f \in \mathcal{F} \setminus \mathcal{N}(\Sigma_n)$. Let $R = (P, Q)$ and $R' = (P', Q')$ be two rational forms with respective degrees (p, q) and (p', q') satisfying $q' \leq 0$ and let $R''' = (P''', Q''')$ their inverse quotient : $R'''^{-1} = R^{-1} \,\%\, R'^{-1}$. Then, if R and R' are two interpolants of f with respect to Σ_n, then R''' is also an interpolant of f with respect to Σ_n.*

3.3 - Reversiblity.

Quotient and inverse quotient are linked by some identities the interest of which is clear although they are not used in the sequel. Let us collect some of them in the following proposition.

Proposition 5.

(i) $$p' \geq 0 \Rightarrow (R \,\%\, R')^{-1} \,\%\, R'^{-1} = \begin{cases} R^{-1} \,\%\, R'^{-1} & \text{if } q' \geq 0 \\ R^{-1} & \text{if } q < q' \end{cases}$$

(ii) $$q' \geq 0 \Rightarrow (R^{-1} \,\%\, R'^{-1}) \,\%\, R' = \begin{cases} R \,\%\, R' & \text{if } p' \geq 0 \\ R & \text{if } p < p' \end{cases}$$

Proof. (i) Thanks to $p' \leq 0$, it is allowed to define $R'' = R \,\%\, R'$. Then $P = AP' + P''$ and $Q = AQ' + Q''$ with $p'' < p'$.

If $q' \geq 0$, then R''' can be defined by $R'''^{-1} = R''^{-1} \ \% \ R'^{-1}$. Then $P'' = BP' + P'''$ and $Q'' = BQ' + Q'''$ with $q''' < q'$. So : $P = (A+B)P' + P'''$ and $Q = (A+B)Q' + Q'''$ with $q''' < q'$. This means $R'''^{-1} = R^{-1} \ \% \ R'^{-1}$ when $q' \geq 0$. Now $q < q'$ implies also $q' \geq 0$, and then $R'''^{-1} = R^{-1} \ \% \ R'^{-1}$. Moreover $q < q'$ implies $R^{-1} \ \% \ R'^{-1} = R^{-1}$ (proposition 3, point iv), so that (i) is proved.

(ii) is proved by applying point (i) to the inverse rational forms.

4 - The Direct Division Algorithm.

4.1 - Definition. *Let $R = (P,Q)$ and $R' = (P',Q')$ be two rational forms with respective degrees (p,q) and (p',q'). For $i = -1,0,1,2,\cdots$ let us define the ordered set of rational forms R_i by :*

$$R_{-1} := R; R_o := R'; k := 0;$$
$$\text{while } p_k \leq 0 \text{ do } R_{k+1} := R_{k-1} \ \% \ R_k; k := k+1 \text{ done.}$$

Let us further refer to this algorithm as the direct form of division algorithm (DDA) working up with initialization (R,R').

4.2 - Foundamental property of DDA.

Proposition 6. *Let a rational interpolation problem with functionals Σ_n satisfying assumptions $(\mathcal{R})$ and $(\mathcal{S})$ on $\mathcal{F}$ and a function $f \in \mathcal{F} \setminus \mathcal{N}(\Sigma_n)$.*

Let (R_i) the ordered set of rational forms obtained by using the DDA with initialization (R,R').

(i) If R and R' are interpolant of f with respect to Σ_n, then each rational form R_i has the same property.

(ii) If $p \geq p'$ and $q \leq q'$ then for every rational form R_k with a non negative index one has : $p_k < p_{k-1}$, $q_k \geq q_{k-1}$ and $p_{k-1} + q_k = p + q'$.

(iii) There is some index $m \leq p'+1$ such that $p_m = -1$.

Proof. (i) is a straightforward consequence of proposition 2 whereas (ii) can be derived from points (ii) and (iii) of proposition 1.

Thanks to the strict decreasing property of p_k, it is possible to bound p_l by $p'-l$ for every index $l \geq 0$ such that R_l is defined. If $R_{p'+1}$ is defined one has $p_{p'+1} \leq -1$, so that $m \leq p'+1$. ∎

4.3 - Justification of the initialization.

Proposition 6 shows us that thanks to a good initialization, the DDA gives rational forms satisfying the interpolation property. The aim of this paragraph is to prove that such an initialization is possible without more restrictive hypothesis.

Proposition 7. *Let a rational interpolation with functional Σ_n satisfying assumptions $(\mathcal{R})$ and $(\mathcal{S})$ on $\mathcal{F}$ and a function $f \in \mathcal{F} \setminus \mathcal{N}(\Sigma_n)$.*

(i) There is a unic polynomial $P_n \in \mathcal{P}_n$ interpolating f with respect to Σ_n.

(ii) There is at least one polynomial $\Pi_n \in \mathcal{P}_{n+1}$ *satisfying :* $\Sigma \in \Sigma_n \Rightarrow \Sigma(\Pi_n) = 0 \wedge \Pi_n \neq 0$. *This polynomial is unic if its coefficient of higher degree is given to be equal to one, in which case* $\Pi_n \in \mathcal{P}^*_{n+1}$.

Proof. (i) Let us define a polynomial P as $P = \Sigma_{j \in I_n} a_j x^j$ with the a_j's in K and x in X, and denote by β_j, $j \in I_n$ a basis of Σ_n. Then the interpolation condition can be written as : $i \in I_n \Rightarrow \Sigma_{j \in I_n} a_j \beta_i(x^j) = \beta_i(f)$. This is a linear system with the matrix $\overset{(n)}{C}$ of hypothesis $(\mathcal{R})$. This assumption insures existence and unicity of the solution. The corresponding polynomial denoted by P_n cannot be zero if $f \notin \mathcal{N}(\Sigma_n)$.

(ii) With similar notations as in (i), the characterisation of Π_n gives : $i \in I_n \Rightarrow \Sigma_{j \in I_{n+1}} a_j \beta_i(x^j) = 0$. For $i \in I_n$ let us define the vector $S_i \in K^{n+2}$, the j$^{\text{th}}$ component of which is $\beta_i(x^j)$. Hypothesis $(\mathcal{R})$ implies that the variety V_n span by these vectors S_n has dimension $n+1$. The equation above means that the vector $a \in K^{n+2}$ with components $a_j, j \in I_{n+1}$ is orthogonal to the variety V_n. Since the vector a cannot be zero (need by $\Pi_n \neq 0$), it is uniquely defined (with respect to a constant factor). Moreover, because it is an interpolant of the zero function, the corresponding polynomial denoted by Π_n cannot be in $\mathcal{P}_n$ without contradiction with (i). ∎

4.4 - How to use the DDA.

Proposition 7 shows there is always an initialization allowing a correct use of the DDA. It is of course a theoretical result and the actual initialization is not obtained by solving a linear system with some general algorithm : P_n can be computed by some polynomial interpolation algorithm whereas Π_n is given by an elementary analysis. For instance, in the case of the modified Hermitian problem (paragraphe 1.4) it is easy to see that $\Pi_n = \prod_{i \in I_n} (x - x_i)^{e_i+1}$. In the special case of Padé approximants, this gives $\Pi_n = x^{n-1}$ and for the Cauchy problem, one has : $\Pi_n = \prod_{i \in I_m} (x - x_i)$.

Proposition 6 is a proof of the DDA : point (i) shows that every form obtained by means of the DDA is a solution of the problem, point (ii) shows that rational forms obtained are ordered with respect of decreasing degree property for numerators and (iii) shows that the algorithm stops.

5 - The inverse division algorithm.

The object of the preceeding algorithm was to give an actual procedure for computing rational interpolants. On the contrary, the aim of he algorithm described below is theoretic : it is only a tool to prove the interest of the first one.

5.1 - Definition and fondamental property of IDA.

Definition. *Let* $R = (P, Q)$ *and* $R' = (P', Q')$ *be two rational forms with respective degrees* (p, q) *and* (p', q'). *For* $i = -1, 0, 1, 2, \ldots$ *let us define the ordered set of rational forms* $\bar{R}_i$ *by*

$\bar{R}_{-1} := R; \bar{R}_0 := R'; k := 0;$

while $\bar{q}_k \leq 0$ do $\bar{R}_{k+1} := (\bar{R}_{k-1} \,\%\, \bar{R}_k^{-1}); k := k + 1$ done.

Let us a refer further to this algorithm as the inverse form of division algorithm (IDA) working up with initialization (R, R').

Proposition 8. *Let a rational interpolation problem with functionals Σ_n satisfying assumptions $(\mathcal{R})$ and $(\mathcal{S})$ on $\mathcal{F}$ and a fonction $f \in \mathcal{F} \setminus \mathcal{N}(\Sigma_n)$.*

Let (R_i) be the ordered set of rational forms obtained by applying the IDA to initialization (R, R').

(i) *If R and R' are two interpolants of f with respect to Σ_n then each rational form $\bar{R}_i$ has the same property.*

(ii) *If $p \leq p'$ and $q \geq q'$ then for every rational form $\bar{R}_k$ with a non negative index, one has : $\bar{q}_k < \bar{q}_{k-1}$, $\bar{p}_k \geq \bar{p}_{k-1}$ and $\bar{q}_{k-1} + \bar{p}_k = q + p'$.*

(iii) *There is some index $\bar{m} \leq q' + 1$ such that $\bar{q}_{\bar{m}} = -1$.*

The proof, closely imitated from the one of proposition 6, is left to the reader.

5.2 - Proposition 9. *Let a rational interpolation problem with functionals Σ_n satisfying assumption $(\mathcal{R})$ and $(\mathcal{S})$ on $\mathcal{F}$ and a function $f \in \mathcal{F} \setminus \mathcal{N}(\Sigma_n)$.*

Let us denote by $(R_k, k \in I'_m)$ the finite sequence of rational forms obtained by using the DDA with initialization (R, R'), where $R = (\Pi_n, 0)$ and $R' = (P_n, 1)$, Π_n and P_n being defined as in proposition 7.

For every rational form R^* interpolating f with respect to Σ_n without being null.

(i) *there is $l \in I'_m$ such that $p_l \leq p^*$ and $q_l \leq q^*$.*

(ii) *there is $A \in \mathcal{P}_{p^*-p_l}$ such that $p^* = AP_l$ and $Q^* = AQ_l$.*

Proof.(i) Let us prove point (i) by reductio ad absurdum.

If (i) is false, there is a rational form R^* interpolating f with respect to Σ_n such that, for every $k \in I'_m$, one has $p_k > p^*$ or $q_k > q^*$. It has been proven in proposition 7 that there is only one rational form interpolating f with respect to Σ_n and satisfying : $p + q \leq n \wedge (q = 0 \vee (q = -1 \wedge p \neq -1))$. Then $q^* \geq 1$, and the set of indexes satisfying $q_k \leq q^*$ is not empty. Let us denote by l the greatest of them : $l = \max\{k \in I_n \mid q_k \leq q^*\}$. One has $l \geq 0$ and the non decreasing property of the q_j's implies $q_l \geq q_o \geq 0$.

From $k \in I_l \Rightarrow q_k \leq q^* \wedge (p_k > p^* \vee q_k > q^*)$ one obtains $p_k > p^*$ for $k \in I_l$. The stopping criterium $p_m = -1$ forbids $p^* < p_m$ and this implies $l < m$. Then the rational form R_{l+1} is defined and it is allowable to write : $0 \leq q_l \leq q^* < q_{l+1}$.

Let us now perform the IDA with the initialisation $\bar{R}_{-1} = R^*$ and $\bar{R}_o = R_l$. One has $\bar{q}_{-1} \geq \bar{q}_o$ and $\bar{p}_{-1} < \bar{p}_o$. Moreover : $\bar{q}_{-1} + \bar{p}_o = q^* + p_l < q_{l+1} + p_l$. From proposition 6, one has $\bar{q}_{l+1} + \bar{p}_l = n - 1$, and then $\bar{q}_{-1} + \bar{p}_o \leq n$. Thanks to proposition 8, one has for $k \in I_{\bar{n}}$: $\bar{p}_k + \bar{q}_{k-1} = \bar{p}_o + \bar{q} = q^* + p_l \leq n$ with $\bar{m} \leq q_l + 1$.

When the IDA stops, the last rational form $\bar{R}_{\bar{m}}$ satisfies $\bar{q}_{\bar{m}} = -1 < \bar{q}_{\bar{m}-1}$ and $\bar{p}_{\bar{m}} + q_{\bar{m}-1} = q^* + p_l$. From $0 \leq q^* + p_l \leq n$ then one has : $0 \leq \bar{p}_{\bar{m}} \leq n$. This fact

contradicts the unicity of the non null rational form interpolating f with respect to Σ_n and satisfying $q = -1$ (proposition 7), then point (i) is proved.

(ii) From (i) there is an index $l \in I'_m$ such that $p_l \leq p^*$ and $q_l \leq q^*$. Let us define a rational form R' by $R'^{-1} = R^{*-1} \,\%\, R_l^{-1}$ Thanks to point (i) of proposition 3, one has $q' < q_l$ and $p' + q_l \leq \max(p_l + q^*, p^* + q_l)$, from which $p' + q_l \leq p^* + q^* \leq n$. Point (ii) of proposition 6 allows us to write $p_{l-1} + q_l = n + 1$. Then $p' + q_l \leq n$ implies : $p' < p_{l-1}$ and the rational form R' satisfies : $p' < p_{l-1}$ and $q' < q_l$. Because inverse quotient respects the interpolation property (proposition 4), R' is an interpolant of f with respect to Σ_n. Proposition 6 implies the monotony of both sequences p_l (decreasing from $n+1$ down to -1) and q_l (increasing from -1 up to q_m) : $k \leq l-1 \Rightarrow p_k \geq p_{l-1} > p'$ and $k \geq l \Rightarrow q_k \geq q_l > q'$. Then the set of indexes j satisfying $p_j \leq p'$ and $q_j \leq q'$ is empty. From point (i) of this proposition : $R' = 0$, and the point (v) of proposition 1 allows us to conclude. Proof of proposition 9 is achieved. ∎

5.3. Corollary *With the same hypothesis and notation as in proposition 9 :*

(i) *If a rational form R interpols $f \in \mathcal{F}\backslash\mathcal{N}(\Sigma_n)$ with respect to Σ_n, there is an index $l \in I_m$ and a polynomial A such that $P = AP_l$ and $Q = AQ_l$.*

(ii) *For every $l \in I_m$ such that P_l and Q_l are not relatively prime, the rational fraction $R' = (P', Q')$ obtained by dividing both P_l and Q_l by their g.c.d. or is not an interpolant of f with respect to Σ_n.*

Proof. Point (i) is a direct formulation of both (i) and (ii) of proposition 9.

In order to prove (ii), let us suppose that the rational form R' interpols f with respect to Σ_n. Because P_l and Q_l are not relatively prime, one has $p < p_l$ and $q < q_l$. By proposition 9, there is an index k such that $R_k = (P_k, Q_k)$ satisfies $P' = AP_k$ and $Q' = AQ_k$. By performing the DDA, there are two indexes l et k satisfying : $p_l > p' \geq p_k$ and $q_l > q' \geq q_k$, and this contradicts the monotony of p_k and q_k (point (ii) of proposition 6). The reduced rational form (confused with a rational fraction) R' is not an interpolant of f with respect to Σ_n. ∎

5.4 - What is brought by this result ?

Before the use of the IDA, it was known that :

1. Thanks to proposition 6, provided that a good initialization is given,

1.1 the DDA gives a sequence of rational forms ordered with respect to the degrees of numerator ;

1.2 each such rational form satisfies the interpolation property with respect to Σ_n ;

1.3 the algorithm stops.

2. Thanks to proposition 7, it is always possible to obtain a good initialization.

The new results given by the use of the IDA are :

1. a positive one : if a rational form R satisfies the interpolation property with respect to Σ_n, this rational form can be reduced to one of the rational forms given

by the DDA ; in other terms, there is no actually different solutions to the initial interpolation problem that rational forms obtained by DDA.

2. a negative one : if a rational interpolant form R_l is not reduced, the corresponding reduced form has no longer the interpolating property.

6. Conclusion

Known as division algorithm, Euclid's algorithm [10] or Kronecker's algorithm, the DDA is not a new one. Its application to more and less difficult problems has been look by Warner [12]. Its reliability has been proved in special cases [2], [4], [10]. What is new can be resumed in the three following points :

1. This algorithm works good when applying to the class of problems characterized by conditions $(\mathcal{R})$ and $(\mathcal{S})$.

2. Every actual solution of the initial problem is given by it (with the exception of an eventual reducing).

3. If a rational form given by the DDA is not reduced, the reduced fraction is no more a solution of the initial problem.

It can be added that the method used here to prove the second and the third results can be used in other cases. For instance similar results have been proved for Baker's algorithm [1] by using a very closed method [5].

However it still remains a negative fact penalizing the use of this elegant algorithm : it gives only rational forms satisfying interpolation conditions. When a reducible rational form is obtained, the rational fraction given by reduction has no more the interpolation property. The problem is to recognize when such a situation occurs. This means that each computed rational form has to be checked in order to know if it is reduced or not. Of course, this situation is not specific to the DDA since it correspond to the discriminating question as posed by Graves-Morris [8] in other rational interpolation algorithms.

References

[1] **G.A. BAKER**
Essentials of Padé Approximants.
Academic Press, New-York, 1975.

[2] **G. CLAESSENS**
Some aspects of the rational Hermite interpolation table and its applications.
Ph. D. Thesis, University of Antwerp (1976).

[3] **G. CLAESSENS**
On the structure of the Newton-Padé Table.
Journal of Approximation Theory, 22 (1978) 304-319.

[4] **F. CORDELLIER**
Deux algorithmes de calcul récursif des éléments d'une table de Padé non normale.
Talk at the *"Colloque sur les approximants de Padé"* Lille 1978.

[5] **F. CORDELLIER**
In progress.

[6] **P.J. DAVIS**
Interpolation and Approximation.
Dover Pub. inc., New-York (1975).

[7] **J. GILEWICZ**
Approximants de Padé.
LN in math. 667 - Springer Verlag, Berlin, Heidelberg, New-York, (1978).

[8] **P.R. GRAVES MORRIS**
The numerical calculation of Padé Approximants.
in *Padé approximants and its applications"*, L. Wuytack ed. LN in Math. 765, Springer Verlag, Berlin, Heidelberg, New-York (1979), pp. 231-245.

[9] **L. KRONECKER**
Zur Theorie des Elimination einer Variabeln aus zwei algebraischen Gleichungen.
Monatsber. K . P. Akad. Wiss. Berlin (1881), p. 535-600.

[10] **J.R. Mc ELIECE and J.B. SHEARER**
A property of Euclid's algorithm and an application to Padé Approximation.
Siam J. Appl. Math. 34 (1978), 611-615.

[11] **J. MEINGUET**
On the solubility of the Cauchy Interpolation Problem.
in *"Approximation Theory"*, A. Talbot ed., Academic Press (1970), 137-163.

[12] **D.D. WARNER**
Hermite interpolation with rational functions.
Ph. D. Thesis, University of California, San Diego (1974).

INTERPRETATION GEOMETRIQUE

DE L' ε-ALGORITHME

F. CORDELLIER

25 Avril 1973

INTRODUCTION

L'ε-algorithme est un algorithme itératif qui, à une suite $(x_n)_{n\in\mathbb{N}}$ de vecteurs de $\mathbb{R}^n$, associe une séquence de suites par un procédé numérique que nous rappelons.

Cet algorithme a été largement étudié par P. Wynn, son auteur, et par C; Brézinski (qui donne en [1] une bibliographie assez complète) : convergence, connexion avec la table de Padé, règles de singularité, etc....

On donne ici une description géométrique d'une itération de l'algorithme. Cette interprétation , qui repose sur la division harmonique, permet de retrouver directement certaines variantes de l'algorithme dans les cas singuliers que Wynn propose en [2] .

1 - Définition algébriques de l'ε -algorithme

1.1 - Domaine de définition

On considère l'espace euclidien $\mathbb{R}^m$ compactifié par l'adjonction du point à l'infini (noté ∞). Cet ensemble est en correspondance biunivoque avec la sphère unité de $\mathbb{R}^{m+1}$ (qu'on notera S_m) par l'inversion de pôle (0, 0,..., 0, 1) et de puissance 2 (qu'on notera), et on le munira de la topologie image par $\mathfrak{j}^{-1}$ de la topologie induite sur cette sphère S_m par la métrique euclidienne de $\mathbb{R}^{m+1}$. L'espace topologique ainsi obtenu sera noté E_m.

Les opérations définies sur $\mathbb{R}^m$ (addition, produit par un réel, et produit scalaire) sont partiellement étendues à E_m par :

$$(1.1.1)\quad \begin{cases} x + \infty = x - \infty = \infty + x = \infty - x = \infty \ , & \forall x \in \mathbb{R}^m \\ \lambda \infty = \infty \ , \ \forall \lambda \neq 0 \end{cases}$$

Par contre, les opérations suivantes restent dépourvues de sens :

$$(1.1.2)\quad \begin{cases} 0 \times \infty \\ \infty + \infty \ , \ \infty - \infty \\ (x/\infty), \ (\infty/x) \ , \ \forall x \in E_m \ , \ \text{où } (x/y) \text{ désigne le produit scalaire dans } \mathbb{R}^m \end{cases}$$

On définit l'opération $y \to y^{-1}$ sur E^m par :

$$(1.1.3)\quad z = y^{-1} \iff z = \begin{cases} y/(y,y) & \text{si } y \in \mathbb{R}^m - \{0\} \\ \infty & \text{si } y = 0 \\ 0 & \text{si } y = \infty \end{cases}$$

Les opérations suivantes

$$(1.1.4)\begin{cases} \mathcal{A}_x(y) = x+y = y+x = y+2 \ , & \forall x \in \mathbb{R}^m \text{ (translation de vecteur x)} \\ \mathcal{H}_{x,\lambda}(y) = x+\lambda\,(y-x) \ , & \forall x \in \mathbb{R}^m, \ \forall \lambda \in \mathbb{R}-\{0\} \\ & \text{(nomothétie de centre x et de rapport } \lambda\text{)} \\ \mathcal{J}_x(y) = x+(y-x)^{-1} \ , & \forall x \in \mathbb{R}^m \\ & \text{(inversion de pôle x et de puissance 1)} \end{cases}$$

sont des fonctions continues de $y \in E_m$ (pour la topologie de E_m)

1.2 - Première définition

Soit $\{x_n\}$ une suite de vecteurs de $\mathbb{R}^m$. On pose :

$$(1.2.1)\begin{cases} \varepsilon_{-1}^{(n)} = 0 \\ \varepsilon_{0}^{(n)} = x_n \end{cases} \qquad n = 0, 1, \ldots$$

et on définit :

$$(1.2.2) \quad \varepsilon_{k+1}^{(n)} = \varepsilon_{k-1}^{(n+1)} + \left(\varepsilon_{k}^{(n+1)} - \varepsilon_{k}^{(n)}\right)^{-1} \qquad \begin{cases} n = 0, 1, \ldots \\ k = 0, 1, \ldots \end{cases}$$

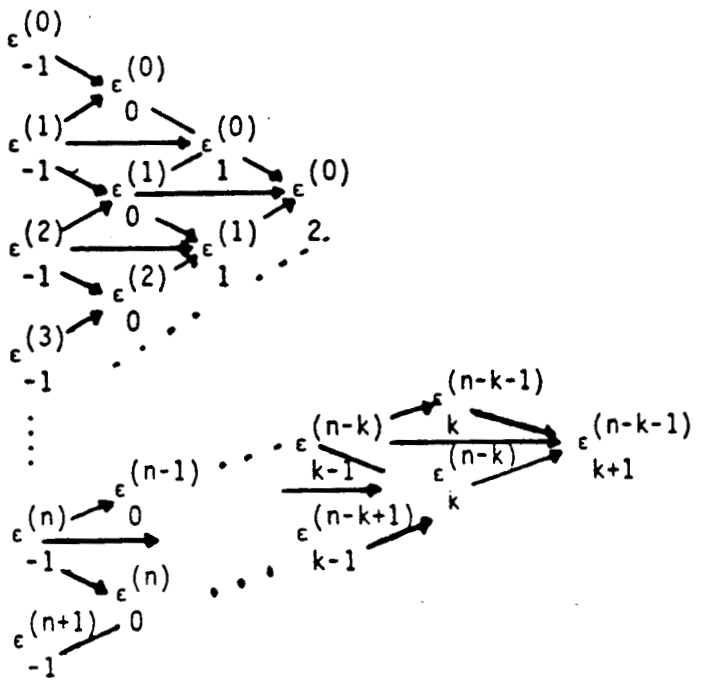
(à condition qu'on n'ait pas simultanément :

$$\varepsilon_{k-1}^{(n+1)} = \infty \qquad \text{ou} \qquad \varepsilon_{k}^{(n+1)} = \varepsilon_{k}^{(n)} = \infty$$

et $\varepsilon_{k}^{(n+1)} = \varepsilon_{k}^{(n)}$

auquel cas l'algorithme sera inapplicable)

L' ε-algorithme consiste alors en la génération du tableau $\varepsilon_k^{(n)}$ suivant :



dans lequel les seules colonnes significatives sont les colonnes d'indices pairs :

$$\varepsilon_0^{(n)} \ , \ \varepsilon_2^{(n)} \ , \ \ldots, \ \varepsilon_{2k}^{(n)} \ , \ldots$$

1.3 - Seconde définition

Il est clair que :

$$(1.3.1) \quad \begin{cases} (y^{-1})^{-1} = y \\ (-y)^{-1} = -(y^{-1}) \end{cases} \qquad \forall y \in E_n$$

La relation (1.2.2) s'écrit alors :

$$(1.3.2) \quad (\varepsilon_{k+1}^{(n)} - \varepsilon_{k-1}^{(n+1)})^{-1} + \varepsilon_k^{(n+1)} = \varepsilon_k^{(n)}$$

Cette même relation (1.2.2) entraine encore :

$$\varepsilon_k^{(n+1)} = \varepsilon_{k-2}^{(n+2)} + (\varepsilon_{k-1}^{(n+2)} - \varepsilon_{k-1}^{(n+1)})^{-1}$$

$$- \varepsilon_k^{(n)} = - \varepsilon_{k-2}^{(n+1)} + (\varepsilon_{k-1}^{(n+1)} - \varepsilon_{k-1}^{(n)})^{-1}$$

$$\varepsilon_{k-2}^{(n+2)} - \varepsilon_{k-2}^{(n+1)} = (\varepsilon_{k-1}^{(n+1)} - \varepsilon_{k-3}^{(n+2)})^{-1}$$

d'où

(1.3.3)
$$\boxed{(\varepsilon_{k+1}^{(n)} - \varepsilon_{k-1}^{(n+1)})^{-1} + (\varepsilon_{k-3}^{(n+2)} - \varepsilon_{k-1}^{(n+1)})^{-1} = (\varepsilon_{k-1}^{(n+2)} - \varepsilon_{k-1}^{(n+1)})^{-1} + (\varepsilon_{k-1}^{(n)} - \varepsilon_{k-1}^{(n+1)})^{-1}}$$
$$n = 0, 1, \ldots$$
$$k = 2, 3, \ldots$$

Cette relation ne fait intervenir que des termes dont les indices de colonne ont la même parité. On peut la définir pour k=1 si on prend la précaution de poser :

(1.3.4)
$$\varepsilon_{-2}^{(n)} = \infty \quad (n = 0, 1, \ldots)$$

Les seules colonnes utiles sont les colonnes d'indice pair. On peut alors définir l'algorithme suivant, où les colonnes d'indice impair sont omises.

On posera :

(1.3.5)
$$\varphi_k^{(n)} = \varepsilon_{2k}^{(n)} \qquad \begin{cases} k = -1, 0, 1, \ldots \\ n = 0, 1, \ldots \end{cases}$$

On initialise :

(1.3.6)
$$\boxed{\left.\begin{aligned} \varphi_{-1}^{(n)} &= \infty \\ \varphi_0^{(n)} &= x_n \end{aligned}\right\} \quad n = 0, 1, \ldots}$$

et on utilise la relation de récurrence :

(1.3.7)
$$\boxed{(\varphi_{k+1}^{(n)} - \varphi_k^{(n+1)})^{-1} + (\varphi_{k+1}^{(n+2)} - \varphi_k^{(n+1)})^{-1} = (\varphi_k^{(n+2)} - \varphi_k^{(n+1)})^{-1} + (\varphi_k^{(n)} - \varphi_k^{(n+1)})^{-1}}$$
$$\begin{cases} k = 0, 1, \ldots \\ n = 0, 1, \ldots \end{cases}$$

2. - Rappels géométriques et notations

2.1. - Vocabulaire

Nous plaçons dans l'espace E_m défini en (1.1.). La correspondance biunivoque de E_m sur S_{m+1} justifie le vocabulaire suivant :

. Toute droite est un cercle passant par le point , et tout plan est une sphère passant par le point ∞.

. Nous dirons que 4 points non confondus x, y, u et v, pris dans cet ordre constituent une division harmonique dans E_m et nous noterons :

(2.1.1) $$(x, y ; u, v) = -1$$

si les 2 conditions suivantes sont remplies :

(2.1.2)
1. ils appartiennent à un même cercle, (non nécessairement unique).
2. leur birapport sur ce cercle est égal à -1.

On traduit encore (2.1.1) en disant que x et y sont conjugués par rapport à u et v, ou que u et v sont conjugués par rapport à x et y.

2.2 - Quelques propriétés

Si 4 points constituent une division harmonique, ils appartiennent à un même cercle, donc à un même plan. Rappelons quelques résultats classiques de la géométrie plane :

Soient 4 points x, y, u et v vérifiant $(x, y ; u, v) = -1$.

(2.2.1) . Si 3 des 4 points sont donnés, alors :

le 4^e est unique $\Leftrightarrow$ les 3 premiers ne sont pas confondus

(2.2.2) . Si 2 des 4 points sont confondus, alors un 3^e est confondu avec les 2 premiers : c'est le seul cas où le cercle défini par les 4 points d'une division harmoniques n'est pas unique.

(2.2.3) . Si l'un des 4 points est le point ∞ , son conjugué est le milieu du segment qui joint les 2 autres points, à condition que ces points soient distincts de ∞. (sinon (2.2.2) est applicable).

Rappelons encore que l'inversion est une transformation géométrique qui conserve :

$$\begin{cases} \text{la nature des cercles et des sphères.} \\ \text{le birapport de 4 points sur un cercle (ou une droite)} \end{cases}$$

Par suite, elle préserve la division harmonique de 4 points.

2.3 - Traduction analytique de la division harmonique

Soient 4 points x, y, u et v tels que : $(x, y ; u, v) = -1$.

Puisque ces 4 points ne sont pas confondus, l'un d'entre eux (au moins) est distinct de ∞. Supposons que ce soit x.

Aux 4 points x, y, u et v, l'inversion J_x (définie en (1.1.4)) associe les points $\bar{x}$, $\bar{y}$, $\bar{u}$ et $\bar{v}$ qui vérifient :

(2.3.1)
$$\begin{cases} \bar{x} = \infty \\ (\bar{x}, \bar{y} ; \bar{u}, \bar{v}) = -1 \end{cases}$$

Or $\bar{y}$, conjugué de ∞ par rapport à $\bar{u}$ et $\bar{v}$, est le milieu du segment $[u, v]$, propriété qui se traduit analytiquement par :

(2.3.2)
$$\bar{y} = \frac{1}{2}(\bar{u} + \bar{v})$$

soit encore :

$$2(\bar{y}-x) = \bar{u}-x+\bar{v}-x$$

En tenant compte de la définition de J_x, il vient :

$$2(y-x)^{-1} = (u-x)^{-1} + (v-x)^{-1}$$

On peut donc écrire :

(2.3.3) $$\left.\begin{array}{l}(x,y\ ;\ u,v) = -1 \\ x \neq \infty\end{array}\right\} \Longrightarrow 2(y-x)^{-1} = (u-x)^{-1} + (v-x)^{-1}$$

Dans le cas $x = \infty$, on a :

(2.3.4) $$(\infty,\ y\ ;\ u,\ v) = -1 \Leftrightarrow 2y = u+v$$

Cette relation étant toujours valable si l'un des 3 points u, v ou y est confondu avec ∞ (puisque $u+\infty = \infty$).

Donc :

(2.3.5) $$(x,\ y\ ;\ u,\ v) = -1 \Leftrightarrow \begin{cases} 2(y-x)^{-1} = (u-x)^{-1} & \text{si } x \neq \infty \\ 2y = u+v & \text{si } x = \infty \end{cases}$$

3. - L'application K : définition et propriétés

3.1 - La fonction H

3.1.1 - Définition et continuité

Définissons formellement les fonctions $h_{x,y}$ et $g_{x,y}$ par :

$$\left.\begin{array}{l} h_{x,y} : u \longrightarrow v \Leftrightarrow (x,y\ ;\ u,v) = -1 \\ g_{x,y} : u \longrightarrow v \Leftrightarrow (x,v\ ;\ u,y) = -1 \end{array}\right| \quad x,y,u, \in E_m$$

la relation (2.3.5) montre que

les fonctions $g_{x,y}$ et $h_{x,y}$ sont continues sur E_m si $x \neq y$ et sur $E - \{x\}$ si $x=y$

Il s'ensuit que :

(3.1.1) la fonction H à valeurs dans E_m, définie formellement sur $E_m \times E_m \times E_m$ par :

$$y = H(x ; u, v) \iff (x, y ; u, v) = -1$$

est définie et continue par rapport à chacun de ses arguments sur $E_m \times E_m \times E_m - \{ x = u = v\}$

(3.1.2) Expression analytique

pourvu que u ou v soit différent de x, on a :

$$H(x ; u, v) = \begin{cases} x + 2\left[(u-x)^{-1} + (v-x)^{-1}\right]^{-1} & \text{si } x \neq \infty \\ \frac{1}{2}(u+v) & \text{si } x = \infty \end{cases}$$

On notera que :

$$H(x ; u, v) = \infty \iff \begin{cases} \text{2 des 3 points x, u, v confondus avec} \infty \\ \underline{\text{ou}} \quad u + v = 2x \end{cases}$$

3.2 - La fonction K

3.2.1 - Définition et continuité

Définissons formellement l'application K de $E_m \times E_m \times E_m \times E_m$ dans E_m par :

(3.2.1.1) $z = K(x ; y ; u, v) = H(y ; x, H(x ; u, v))$

soit $z = H(y ; x, u, w)$ où $w = H(x ; u, v)$

K sera continue (par rapport à chaque variable) si :

1. $w = H(x ; u, v)$ est définie (donc continue)

   $\Leftrightarrow \quad (x ; u, v) \in E_m^3 - \{u = v = x\}$

2. $z = H(y ; x, w)$ est définie (donc continue)

   $\Leftrightarrow \quad (y ; x, w) \in E_m^3 - \{y = x = w\}$

or $w = x \Leftrightarrow x = u$ ou $v = x$

(3.2.1.2) K est donc définie et continue sur $E_m \times E_m \times E_m \times E_m$ privé de :

$$\{x = u = v\} \cup \{x = v = y\} \cup \{x = y = u\}$$

3.2.2 - Expression analytique

. Supposons $x \neq \infty$.

$w = H(x ; u, v) \Leftrightarrow 2(w-x)^{-1} = (u-x)^{-1} + (v-x)^{-1}$

$z = H(y ; x, w) \Leftrightarrow 2(w-x)^{-1} = (y-x)^{-1} + (z-x)^{-1}$

d'où

$$(u-x)^{-1} + (v-x)^{-1} = (y-x)^{-1} + (z-x)^{-1}$$

. supposons $x = \infty$

$w = H(\infty ; u, v) = \frac{1}{2}(u+v)$

$z = H(y ; \infty, w) \Rightarrow w = \frac{1}{2}(y+z)$

d'où $z = u+v-y$

(3.2.2.1)

$$K(x ; y ; u, v) = \begin{cases} x + ((u-x)^{-1} + (v-x)^{-1} - (y-x)^{-1})^{-1} & \text{si } x \neq \infty \\ u + v - y & \text{si } x = \infty \end{cases}$$

3.3 - Propriétés géométriques de l'application K.

3.1.1 - Définition géométrique directe

par définition :

$$y = K(x\ ;\ y\ ;\ u,\ v) \iff \begin{cases} (x,\ w\ ;\ y,\ z) = -1 \\ \text{où}\quad w : (x,\ w\ ;\ u,\ v) = -1 \end{cases}$$

Si C est un cercle passant par les 3 points x, u et v, alors w est le conjugué de x par rapport à u et v sur C.
Si Γ est un cercle passant par les 3 points x, w et y, alors z est le conjugué de y par rapport à x et w sur Γ.

Remarques 1. Le cercle C est unique si et seulement si les 3 points x, u et v sont distincts. Dans le cas contraire, w est confondu avec le point double (pas de point triple, par définition) : w est donc indépendant du choix de C.

2. La même remarque s'applique au cercle Γ.

Localisation directe de z :

Toute sphère contenant les 4 points x, u, v et z contient un cercle C, donc le point w. Elle contient donc un cercle Γ, donc le point z.
Si les points x, u, v et z ne sont pas cocycliques, cette sphère est unique.
Le même raisonnement montre que si les points x, u, v et z sont cocycliques, alors z appartient à tout cercle passant par ces points.

Si le cercle défini par x, u, v et z n'est pas unique, alors z est confondu avec l'un des 2 points qui constituent leur intersection.

3.3.2 - L'inversion $\mathcal{J}_x$ ($x \neq \infty$)

Soient 5 points x, u, v, u et z $\in E_m$ et vérifiant :

$$\begin{cases} z = K(x\ ;\ y\ ;\ u,\ v) \\ x \neq \infty \end{cases}$$

C'est-à-dire qu'il existe w $\in E_m$ tel que :

$$\begin{cases} (x,\ w\ ;\ u,\ v) = -1 \\ (x,\ w\ ;\ y,\ z) = -1 \\ x \neq \infty \end{cases}$$

Notons $\bar{x}$, $\bar{u}$, $\bar{v}$, $\bar{y}$, $\bar{z}$ et $\bar{w}$ les inverses respectifs de x, u, y; z et w dans l'inversion $\mathcal{J}_x$. Ils vérifient :

$$\begin{cases} (\bar{x},\ \bar{w}\ ;\ \bar{u},\ \bar{v}) = -1 \\ (\bar{x},\ \bar{w}\ ;\ \bar{y},\ \bar{z}) = -1 \\ \bar{x} = \infty \end{cases}$$

Ceci se traduit pour :

$$2\bar{w} = \bar{u} + \bar{v} = \bar{y} + \bar{z}$$

w est le milieu commun des segments $[\bar{u},\bar{v}]$ et $[\bar{y},\bar{z}]$

3.3.3 - Cercles tangents

Supposons toujours que $x \neq \infty$, avec, en plus :

Les points x, u, v et y ne sont pas cocycliques.

Alors ces points définissent une sphère S et les points $\bar{x}$, $\bar{u}$, $\bar{v}$, $\bar{y}$, $\bar{z}$ et $\bar{w}$ appartiennent au plan $\bar{S}$ (inverse de la sphère S) parallèle au plan T_x tangent

en x à la sphère S.

Puisque $\bar{w}$ est le milieu des segments $[\bar{u},\bar{v}]$ et $[\bar{y},\bar{z}]$, le quadrilatère [u, y, v, z] est un parallèlogramme, d'où :

$$[u, y] \;//\; [v, z]$$
$$[u, z] \;//\; [v, y]$$

Les inverses des droites qui portent ces segments sont des cercles passant par x, et le parallélisme de 2 droites équivaut à la tangence en x de leurs inverses. Donc :

$$(3.3.3.1)\quad \begin{cases} \text{les cercles } (x, u, y) \text{ et } (x, v, z) \text{ sont tangents en } x. \\ \text{les cercles } (x, u, z) \text{ et } (x, y, v) \text{ sont tangents en } x. \end{cases}$$

Puisque les triangles $(\bar{u}, \bar{y}, \bar{w})$ et $(\bar{v}, \bar{z}, \bar{w})$ se correspondent $\bar{w}$, leurs cercles circonscrits aussi dans la symétrie de centre .

Ces cercles sont tangents en w et cette propriété se conserve dans l'inversion $\mathfrak{I}_x$.

La même remarque s'applique aux triangles $(\bar{u}, \bar{z}, \bar{w})$ et $(\bar{v}, \bar{y}, \bar{w})$.

Donc :

$$(3.3.3.2)\quad \begin{cases} \text{les cercles } (u, y, w) \text{ et } (v, z, w) \text{ sont tangents en } w \\ \text{les cercles } (u, z, w) \text{ et } (v, y, w) \text{ sont tangents en } w \end{cases}$$

3.3.4 - Autres divisions harmoniques

Supposons $x \neq \infty$. Reprenons les notations de (3.3.2).

Notons $\bar{\alpha}$ et $\bar{\beta}$ les symétriques de y par rapport à respectivement $\bar{v}$ et $\bar{u}$.

Alors $\bar{z}$ est le milieu de $[\bar{\alpha}, \bar{\beta}]$.

En passant aux inverses α et β de $\bar{\alpha}$ et $\bar{\beta}$, on obtient :

(3.3.4.1) { si α est le conjugué de y par rapport à x et v et β celui de y par rapport à x et u alors z est celui de x par rapport à α et β .

Soit $\bar{\lambda}$ le symétrique de v par rapport à $\bar{y}$, et $\bar{\sigma}$ le milieu de $[\bar{y}, \bar{u}]$; alors $\bar{\sigma}$ est le milieu de $[\bar{z}, \bar{\lambda}]$.

En passant aux inverses λ et σ de $\bar{\lambda}$ et $\bar{\sigma}$, il vient :

(3.3.4.2) { Si λ est le conjugué pe v par rapport à x et y et σ celui de x par rapport à u et y, alors z est celui de λ par rapport à x et σ .

Remarque : L'examen du cas $x = \infty$ montre que les propriétés des paragraphes (3.3.3) et (3.3.4) sont conservées.

3.3.5 - Déterminations géométriques de $z = K(x ; y; u, v)$

. Rappelons pour mémoire la construction (3.1.1), possible dans tous les cas où K est définie .

. Les propriétés (3.3.4) fournissent des constructions analogues valables dès que K est définie. Elles offrent l'inconvénient de faire intervenir une division harmonique supplémentaire.

. La propriété (3.3.3.1) suppose que les points x, u, v et y ne soient pas cocycliques. On a alors :

Si on note C_1 le cercle (x, u, y) et C_2 le cercle (x, y, v), alors z est l'intersection du cercle passant par v et tangent en x à C_1 avec le cercle passant par u et tangent en x à C_2.

Remarque : l'application K se prête à d'autres interprétations géométriques. Toutefois, ces interprétations sont le plus souvent liées à la disposition relative des points x, u, v et y : (coplanarité, cocircularité, alignement, etc...) et leur portée s'en trouve très affaiblie. Il en est de même de K.

On peut signaler par exemple cette caractérisation de $w = H(x ; u, v)$ quand les points x, u et v ne sont pas alignés :

w est le point dont la projection sur [u, v] est le milieu de ses projections sur [x, u] et [x, v] .

Cette caractérisation d'une division harmonique sur un cercle n'a pas d'équivalent pour des points alignés.

4. - Définition géométrique de l' ε-algorithme

4.1. - Définition formelle

Compte-tenu de (3.2) on peut encore définir l'algorithme (1.3) par :

$$(1.3.6)\qquad \left.\begin{aligned} \varphi_{-1}^{(n)} &= \infty \\ \varphi_{0}^{(n)} &= x_n \end{aligned}\right\} \quad n = 0, 1, \ldots$$

$$(4.1.1)\qquad \varphi_{k+1}^{(n)} = K(\varphi_{k}^{(n+1)} ; \varphi_{k-1}^{(n+2)} ; \varphi_{k}^{(n)} , \varphi_{k}^{(n+1)}) \qquad \begin{matrix} k = 0, 1, \ldots \\ n = 0, 1, \ldots \end{matrix}$$

à condition que à chaque étape, les 2 conditions suivantes soient satisfaites:

$$(4.1.2)\qquad . \quad \varphi_{k}^{(n+1)} \neq \infty$$

(4.1.3) . au plus un des 3 points $\varphi_{k-1}^{(n+2)}$, $\varphi_{k}^{(n)}$, $\varphi_{k}^{(n+2)}$ est confondu avec $\varphi_{k}^{(n+1)}$.

Nous avons vu que, pourvu que la condition (4.1.3) soit satisfaite, la fonction K reste continue au voisinage de $\varphi_k^{(n+1)} = \infty$

Nous pouvons donc prolonger l'algorithme par continuité (dans E_m) en éliminant la condition (4.1.2).

4.2 - Interprétation géométrique

Nous pouvons prêter à K les diverses interprétations du paragraphe (3.4).

En particulier :

1) moyennant la condition (4.1.3) on définit : $\gamma_k^{(n+1)}$ par :

$(\varphi_k^{(n)}, \varphi_k^{(n+2)}; \varphi_k^{(n+1)}, \gamma_k^{(n+1)}) = -1$

alors $\varphi_{k+1}^{(n)}$ vérifie : $(\varphi_k^{(n+1)}, \gamma_k^{(n+1)}; \varphi_{k-1}^{(n+2)}, \varphi_{k+1}^{(n)}) = -1$

2) moyennant la condition (4.1.3) on définit $\alpha_k^{(n+1)}$ et $\beta_k^{(n+1)}$ par :

$(\varphi_k^{(n+1)}, \varphi_{k-1}^{(n+2)}; \varphi_k^{(n)}, \alpha_k^{(n+1)}) = -1$

$(\varphi_k^{(n+1)}, \varphi_{k-1}^{(n+2)}; \varphi_k^{(n+2)}, \beta_k^{(n+1)}) = -1$

alors $\varphi_{k+1}^{(n)}$ vérifie : $(\alpha_k^{(n+1)}, \beta_k^{(n+1)}; \varphi_k^{(n+1)}, \varphi_{k+1}^{(n)}) = -1$

4.3. - Expression analytique

D'après (3.2.2), on a :

$$\varphi_{k+1}^{(n)} = \begin{cases} \varphi_k^{(n+1)} + \left[(\varphi_k^{(n+2)} - \varphi_k^{(n+1)})^{-1} + (\varphi_k^{(n)} - \varphi_k^{(n+1)})^{-1} - (\varphi_k^{(n+2)} - \varphi_k^{(n+1)})^{-1}\right]^{-1} & \text{si } \varepsilon_k^{(n+1)} \neq \infty \\ \varphi_k^{(n)} + \varphi_k^{(n+2)} - \varphi_{k-1}^{(n+2)} & \text{sinon} \end{cases}$$

Nous retrouvons ici la règle singulière proprosée par Wynn [2] dans le cas où un terme d'une colonne d'indice pair devient infini.

4.4 - Les limites de l'interprétation géométrique simple

Supposons que 2 termes consécutifs $\varphi_k^{(n)}$ et $\varphi_k^{(n+1)}$ soient égaux, alors que les termes $\varphi_k^{(n-1)}$, $\varphi_k^{(n+2)}$, $\varphi_{k-1}^{(n+1)}$ et $\varphi_k^{(n+2)}$ sont distincts de leur valeur commune x.

La forme géométrique de l'ε-algorithme permet de définir :

$$\varphi_{k+1}^{(n-1)} = K(\varphi_k^{(n)} ; \varphi_{k-1}^{(n+1)} ; \varphi_k^{(n-1)}, \varphi_k^{(n+1)})$$

$$\varphi_{k+1}^{(n)} = K(\varphi_k^{(n+1)} ; \varphi_{k-1}^{(n)} ; \varphi_k^{(n)}, \varphi_k^{(n+2)})$$

et on vérifie aisément que $\varphi_k^{(n)} = \varphi_k^{(n+1)} = x \Rightarrow \varphi_{k+1}^{(n-1)} = \varphi_{k+1}^{(n)} = x$

Il s'ensuit que :

$\varphi_{k+2}^{(n-1)} = K(\varphi_{k+1}^{(n)} ; \varphi_k^{(n+1)} ; \varphi_{k+1}^{(n-1)}, \varphi_{k+1}^{(n+1)})$ est indéfini.

et le déroulement (3 points confondus) de l'algorithme est interrompu.

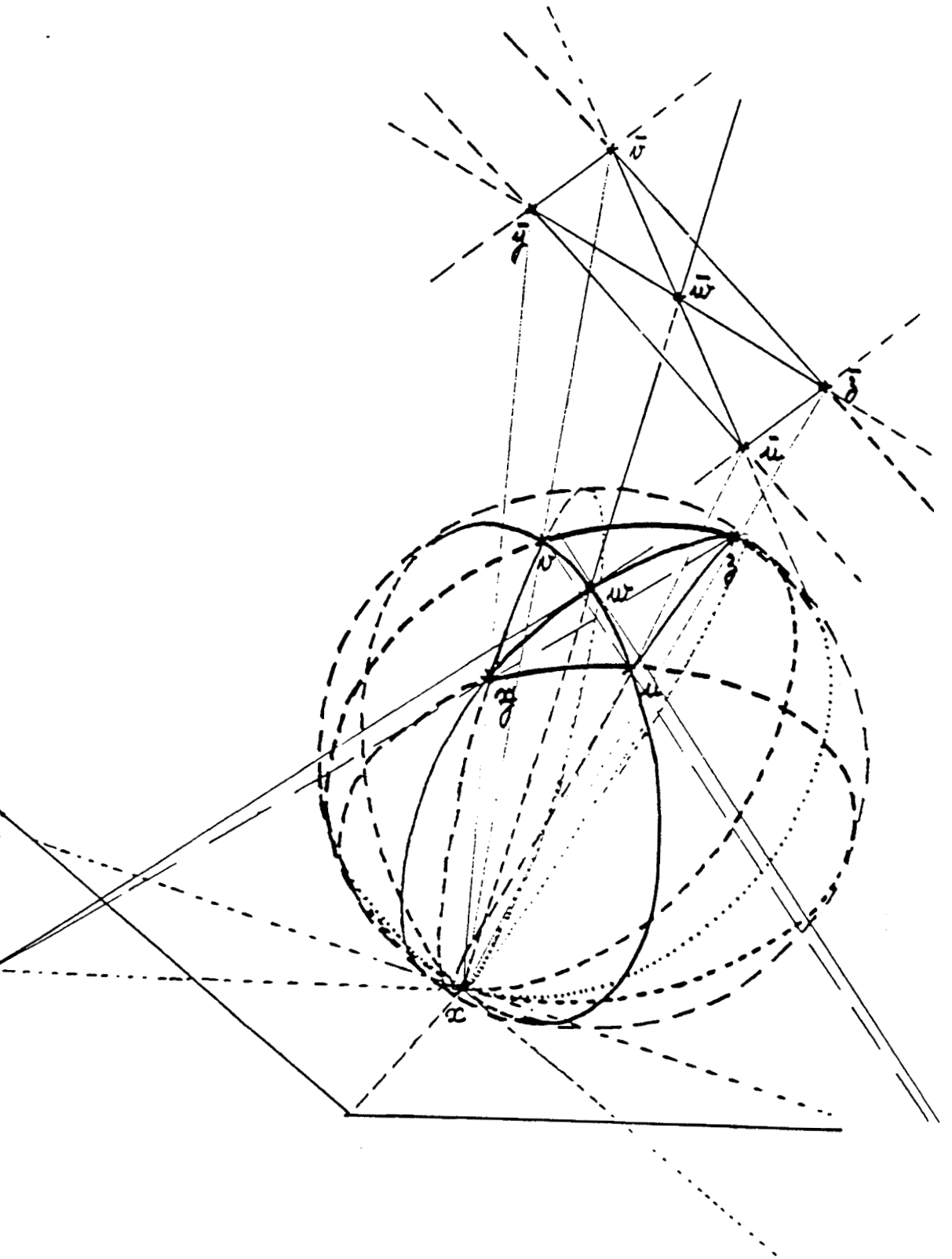
L'interprétation géométrique ne nous permet pas de retrouver directement la seconde règle singulière de Wynn. (2 points consécutifs confondus).

En fait, nous montrerons dans un prochain papier que cette interprétation peut être étendue à un algorithme qui reste applicable lorsque p termes $\varphi_k^{(n+i)}$ (i=0,...,p-1) sont confondus, ce qui généralise les règles singulières de [2].

REFERENCES

(1) C. BREZINSKI Méthodes d'accélération de la convergence en Analyse Numérique
Thèse - Grenoble - (1971)

(2) P. WYNN Singular rules for certain non-linear algorithms
BIT - 3 - (1963) - pp. 175-195.



Règles singulières pour l'ε-algorithme vectoriel

Résumé: l'ε-algorithme vectoriel est un algorithme de transformation non linéaire de suites qui consiste à construire un tableau colonne par colonne. Sa mise en œuvre directe est impossible lorsque plus de deux termes consécutifs d'une même colonne sont égaux. On propose ici une nouvelle forme de cet algorithme qui permet de le mettre en œuvre quelle que soit la suite initiale et qu'on obtient en utilisant les propriétés de continuité que présente cette transformation.

Note: rédigé en juillet 1976, ce papier développe un travail ébauché en 1974, travail dont j'avais présenté les grandes lignes au Colloque National d'Analyse Numérique de Gourette en juin 1974. En annexe figurent un programme mettant en œuvre ces règles singulières et le schéma de la démonstration permettant l'identité de Wynn à une table de Padé non normale, conséquence du présent travail. Une preuve plus directe de ce dernier résultat a été publiée en 1979. Le récent travail de P. Graves-Morris et C.D. Jenkins étend à la table des GIPAs (généralisation vectorielle des approximants de Padé) la généralisation de l'identité de Wynn que j'ai établi ici pour l'ε-agorithme par des voies bien différentes.

Introduction

L'ε-algorithme vectoriel est un algorithme de transformation non linéaire de suites de vecteurs à composantes réelles ou complexes proposé par Wynn [18,20], algorithme dont la principale application est l'accélération de la convergence de suites obtenues en utilisant des méthodes itératives, mais qu'on peut utiliser à d'autres fins, comme par exemple, la résolution de systèmes d'équations linéaires ou non [3], la détermination des éléments propres d'une matrice [5] ou l'estimation des paramètres dans les modèles mathématiques [23]. Il con-siste essentiellement en la génération d'un tableau triangulaire construit colonne par colonne de la gauche vers la droite et de haut en bas au moyen d'une simple règle de calcul. Il convient toutefois de noter que cette règle de calcul est en défaut lorsque plusieurs valeurs consécutives d'une même colonne de ce tableau sont égales, mais Wynn [21] a proposé une règle singulière permettant de traiter le cas où deux valeurs consécutives sont confondues dans le cas de l'ε-algorithme scalaire, et l'auteur [8] a récemment proposé une règle semblable pour le cas vectoriel. Néanmoins, l'algorithme reste indéfini dans le cas où plus de deux valeurs consécutives d'une même colonne sont identiques.

L'objet de ce papier est de lever cette indétermination. Pour cela, nous proposons un algorithme qui consiste, lui aussi, à générer un tableau triangulaire dans des conditions similaires à celles de l'ε-algorithme vectoriel classique, mais avec des règles plus élaborées. Comme nous le verrons, la construction de ce tableau est toujours possible, quelle que soit la suite initiale, et cette construction s'identifie avec celle de l'ε-algorithme vectoriel classique dans le cas où la mise en œuvre de ce dernier est possible: il s'agit donc bien d'une extention de la définition de l'ε-algorithme vectoriel. La méthode suivie pour aboutir à la définition de cet algorithme est une méthode de prolongement par continuité de la définition de l'ε-algorithme vectoriel ou, si on préfère, de la transformation non linéaire de suites que sa mise en œuvre représente. Il convient de préciser que le tableau que nous construisons ainsi à partir d'une suite donnée peut comporter des "trous", c'est à dire qu'il existe des indices pour lesquels la valeur de ce tableau n'est pas définie. Ces valeurs sont localisées dans des blocs carrés, et elles sont environnées par des valeurs égales. La structure par blocs carrés des tables de Padé non normales [11] se retrouve donc dans le tableau associé à la mise en œuvre de l'ε-algorithme vectoriel, ce que permettait d'espérer la connexion entre la table de Padé et le tableau associé à la mise en œuvre de l'ε-algorithme scalaire [19].

Pour illustrer l'intérêt que présente cette extension de l'ε-algorithme, plaçons-nous tout d'abord dans le cas de l'ε-algorithme scalaire. Nous savons alors que tout terme du tableau associé est le quotient de deux déterminants de Hankel construits à partir de la suite initiale [14]. L'ε-algorithme consiste à utiliser des relations récurrentes entre ces quotients de déterminants pour pouvoir calculer ces quotients sans devoir calculer les déterminants eux-mêmes. Un terme indéfini du tableau correspond au quotient de deux déterminants nuls. Alors que la présence d'un terme indéfini dans le tableau interdit la poursuite de l'ε-algorithme, elle ne nuit en rien à la mise en œuvre de l'algorithme que nous proposons, ce qui permet de calculer tous les termes définis du tableau, quelle que soit la suite initiale. Par la même occasion, cet algorithme autorise la construction de la table de Padé [11] alors que l'ε-algorithme classique est en défaut pour les tables de Padé non normales. Dans le cas de l'ε-algorithme vectoriel, nous ignorons ce que représentent les termes du tableau construit à partir de la suite initiale. Les résultats théoriques [4,16] dont nous disposons sont trop fragmentaires pour mettre en lumière la nature exacte de la transformation de suites que représente cet algorithme. L'algorithme proposé est alors un simple prolongement par continuité de la transformation de suites associée à l'ε-algorithme vectoriel.

Annexe4: Règles singulières pour l'ε-algorithme vectoriel.

Dans la première partie de ce travail, nous proposerons une interprétation géométrique élémentaire d'une étape de l'ε-algorithme vectoriel. Nous ne développerons pas ici l'aspect géométrique de cette interprétation que nous avons déjà abordé en [7], mais nous utiliserons cette interprétation pour établir la continuité d'un élément du tableau en fonction des éléments des colonnes antérieures pour la topologie de l'espace compactifié. Un premier paragraphe est consacré au rappel des quelques concepts topologiques ou géométriques dont nous avons besoin dans le second paragraphe où nous étudions l'application K qui intervient dans la définition de l'ε-algorithme vectoriel. Un troisième paragraphe se rapporte à diverses définitions de l'ε-algorithme vectoriel parmi lesquelles on trouvera une définition géométrique.

Dans la seconde partie, nous utilisons la continuité d'un élément du tableau en fonction de ceux des colonnes antérieures pour étendre la définition de l'ε-algorithme vectoriel. l'idée de base de cette étude est de considérer l'ensemble des valeurs qui interviennent dans le tableau associé à la mise en œuvre de l'algorithme comme un être mathématique indépendant qu'on appellera ε-tableau. Comme il sera utile, et même indispensable, de travailler sur les tableaux associés aux deux formes de l'ε-algorithme, la forme en croix et la forme en losange, c'est deux formes d'ε-tableaux que nous introduirons: les ε-tableaux en losanges et les ε-tableaux en croix. Ces ε-tableaux sont des applications définies sur des ensembles de points comportant des losanges (ou des croix) et ils doivent vérifier des conditions qui lient leurs valeurs sur un losange (ou une croix). Cela nécessite l'introduction d'ensembles structurés sur lesquels on puisse les définir: ce sont les quadrillages en quinconce (comportant des losanges) et les quadrillages en grille (comportant des croix). En fait, compte tenu de la relation du losange (ou de la croix) qui lie les valeurs définies en cinq points voisins du quadrillage, un ε-tableau sera défini dès qu'on connait ses valeurs sur un certain sous-ensemble de points du quadrillage. Un tel ensemble sera appelé base du quadrillage. Il peut correspondre à l'initialisation de l'ε-algorithme.

Toutefois, la donnée des valeurs d'un ε-tableau sur les points d'une base de quadrillage ne permet pas toujours de définir cet ε-tableau sur le quadrillage tout entier. Si c'est possible, nous dirons que l'ε-tableau est normal: l'ε-algorithme classique ne nous permet de construire un ε-tableau que s'il est normal. Si ce n'est pas le cas, la partie du quadrillage sur laquelle l'ε-tableau peut être défini sera appelée socle de cet ε-tableau. L'ensemble des ε-tableaux définis sur une base de quadrillage peut être muni d'une structure topologique. Un des résultats essentiels que nous établirons est la densité des ε-tableaux normaux dans cette structure. L'introduction d'une nouvelle notion, celle d'ε-tableaux quasi-normaux, nous permet d'affirmer que cette classe d'ε-tableaux réalise le prolongement par continuité des ε-tableaux normaux pourvu que les valeurs sur la base correspondent aux conditions d'initialisation classiques de l'ε-algorithme.

Il est clair que le résultat obtenu est le fruit de l'introduction de concepts nouveaux qui, bien qu'élémentaires, doivent être présentés avec précision. Le quatrième paragraphe est consacré à la définition des quadrillages et le cinquième à celle des ε-tableaux. Le sixième paragraphe établit certains résultats relatifs aux limites d'ε-tableaux normaux tandis que le septième et dernier présente les ε-tableaux quasi-normaux et permet alors de définir l'extension de l'ε-algorithme vectoriel.

En annexe, on trouvera deux résultats que je n'ai pas pris soin de développer davantage, mais dont l'intérêt n'est guère discutable: le premier est un programme Pascal mettant en œuvre les règles singulières établies dans ce papier, et le second est un schéma de démonstration de l'extension de l'identité de Wynn aux tables de Padé non normales. Ce programme Pascal est le transcodage d'un programme initialement écrit en Algol; celui du second pâtit du fait que j'ai présenté ultérieurement une démonstration totalement algébrique de l'identité généralisée de Wynn [9], et que Graves-Morris et Hopkins [12,13] l'ont récemment étendue aux GIPAs.

1. Rappels géométriques

1.1 L'espace E_m.

1.1.1 définition.

On considère l'espace euclidien $\mathbb{R}^m$ compactifié par l'adjonction du point à l'infini noté ∞. L'espace topologique E_m ainsi obtenu est en correspondance biunivoque avec la sphère unité S_m de $\mathbb{R}^{m+1}$ dans l'inversion de pôle $(0,\dots,0,1)$ et de puissance 2 (notée φ), et cette correspondance respecte les topologies respectives de E_m et de S_m en ce sens que la topologie de E_m n'est autre que la topologie image par $\varphi^{-1}=\varphi$ de la topologie induite sur la sphère S_m par la métrique euclidienne de $\mathbb{R}^{m+1}$. Il est clair que E_m n'est autre que l'**espace anallagmatique réel** étudié dans le cas m=3 par Hadamard [15] dans le cadre de la géométrie anallagmatique.

1.1.2 Opérations sur E_m.

Les opérations définies sur $\mathbb{R}^m$ sont partiellement étendues à E_m par:

$$(1.1)\quad \begin{cases} x+\infty=\infty+x=x-\infty=\infty-x=\infty & \forall x \in \mathbb{R}^m \\ \lambda\infty=\infty & \forall \lambda \in \mathbb{R}\setminus\{0\} \end{cases}$$

Par contre, les opérations suivantes restent dépourvues de sens:

$$(1.2)\quad \begin{cases} 0*\infty \quad \infty+\infty \quad \infty-\infty \\ \langle\infty|x\rangle \quad \langle x|\infty\rangle \quad \forall x \in E_m \end{cases}$$

où $\langle x|y\rangle$ représente le produit scalaire dans $\mathbb{R}^m$.

Définissons l'opération : $y \to y^{-1}$ de E_m dans E_m par:

$$(1.3)\qquad z=y^{-1}=\begin{cases} y/\langle y|y\rangle & \text{si } y \in \mathbb{R}\setminus\{0\} \\ \infty & \text{si } y=0 \\ 0 & \text{si } y=\infty \end{cases}$$

z est l'inverse de Samelson du vecteur y [24], c'est à dire l'inverse géométrique de y dans l'inversion de pôle l'origine et de puissance l'unité.

Les opérations suivantes:

$A(x,y) = x+y=y+x$ (translation)

$H_\lambda(x,y) = x+\lambda*(y-x), \quad \forall \lambda \in \mathbb{R}$ (homothétie)

$J(x,y) = x+(y-x)^{-1}$ (inversion)

définissent des applications continues de $(x,y) \in \mathbb{R}^m \times E_m$ dans E_m.

Nous noterons I_x l'inversion de pôle x et de puissance 1:

$$I_x(y) = J(x,y), \quad \forall x \in E_m.$$

Remarque: E_m n'est évidemment pas un espace vectoriel, et les règles de calcul (1.1) ne doivent pas masquer l'éventualité d'opérations sans résultat (1.2). En particulier, les règles courantes utilisées dans la résolution des équations doivent être maniées avec précaution: c'est ainsi que l'ajout d'un même terme aux deux membres d'une équation ne fournit une équation équivalente que si ce terme est distinct de ∞. Le changement de membre qui en est un cas particulier comporte la même restriction.

1.1.3 Espaces complexes.

L'espace vectoriel $\mathbb{C}^m$ est un espace de dimension 2m sur $\mathbb{R}$ sur lequel on peut définir le produit scalaire réel:

$$(1.4) \qquad \langle x|y\rangle = \sum_{i=1}^{m} (x_i\bar{y}_i + \bar{x}_i y_i)/2 \quad (\text{ en notant } \bar{t} \text{ le conjugué de t dans } \mathbb{C}),$$

qui en fait un espace euclidien isomorphe à $\mathbb{R}^{2m}$. Sa compactification définit un espace topologique F_m isomorphe à E_{2m} pourvu qu'on définisse des opérations analogues à (1.1). Dans cet cet isomorphisme, l'opération (1.3) dans E_{2m} correspond à: $y^{-1}=y/\langle y,y\rangle$ si $y \in \mathbb{C}^m-\{0\}$.

Cet espace est distinct de l'espace hermitien $\mathbb{C}^m$ où le produit scalaire est défini par:

$$(1.5) \qquad \langle\langle x|y\rangle\rangle = \sum_{i=1}^{m} x_i\bar{y}_i.$$

Dans l'espace hermitien compactifié (noté G_m), l'inverse de Samelson vérifie:

$$(1.6) \qquad y^{-1}=\bar{y}/\langle\langle y,y\rangle\rangle \text{ si } y \in \mathbb{C}^m\setminus\{0\}.$$

Puisque $\langle\langle y|y\rangle\rangle = \langle y|y\rangle$, l'inverse dans G_m est le conjugué de celui que nous avons défini dans F_m et qui respecte l'isomorphisme de F_m et E_{2m}. En fait, il est indifférent de choisir l'un ou l'autre de ces inverses dans la définition de l'ε-algorithme vectoriel complexe (proposition 3.1). Il s'ensuit que la mise en œuvre de l'ε-algorithme dans l'espace complexe compactifié F_m (ou G_m) est équivalente à celle de l'ε-algorithme dans E_{2m}. On ne restreint donc pas la généralité en se limitant aux espaces réels, ce que nous ferons désormais.

1.2 Rappels de géométrie.

1.2.1 Cercles et sphères.

En accord avec la terminologie classique, nous appelons droite et plan les variétés linéaires de $\mathbb{R}^m$ de dimension respectives 1 et 2. L'adjonction du point à l'infini compactifie, non seulement l'espace $\mathbb{R}^m$, mais aussi tous ses sous-espaces et les variétés linéaires qui s'en déduisent par translation: le point à l'infini est commun à toutes les variétés linaires compactifiées de E_m, donc aux droites et aux plans compactifiés. Dans un souci d'unité, nous considérerons les droites (respectivement: les plans) comme des cercles (respectivement: des sphères) contenant le point à l'infini de E_m. Ceci revient à adopter pour l'ensemble E_m le vocabulaire que nous utiliserions sur la sphère $S_m=\varphi(E_m)$, c'est à dire celui de la géométrie anallagmatique [15].

1.2.2 Division harmonique.

La notion de division harmonique est suffisamment classique pour ne pas être rappelée ici. Dans ce papier, son usage est restreint au cas des points cocycliques ou alignés. Nous dirons que quatre points a,b,c et d, pris dans cet ordre, forment une **division harmonique** si les deux conditions suivantes sont remplies:
1) ils appartiennent à un même cercle;
2) leur birapport (a,b;c,d) est égal à -1.

Compte tenu de la définition restrictive que nous retenons, nous nous contentons de l'écriture (a,b;c,d)=-1 pour traduire ces deux propriétés. En vertu de la symétrie des rôles joués par les deux paires de points (a,b) et (c,d), nous dirons indifféremment que les points d'un couple sont **conjugués** par rapport à ceux de l'autre.

Si quatre points a,b,c,d constituent une division harmonique, ils appartiennent à un même cercle, donc à un même plan. Rappelons donc quelques résultats classiques de la géométrie plane:

*Si trois des quatre points sont donnés, le quatrième est unique si et seulement si les trois autres ne sont pas confondus.
*Si deux des quatre points sont confondus, alors le point double est triple.
*Si l'un des quatre points est le point à l'infini, son conjugué par rapport à deux points distincts de l'infini est le milieu du segment qui joint ces deux points.

Proposition1: *une condition nécessaire et suffisante pour que (a,b;c,d)=-1 est que l'une des trois conditions suivantes soit remplie:*
1) a=c=d,
2) a≠∞ et $2(b-a)^{-1}=(c-a)^{-1}+(d-a)^{-1}$,
3) a=∞ et 2b=c+d.

1.2.3 Géométrie anallagmatique.

La géométrie anallagmatique est l'étude des propriétés géométriques invariantes par inversion. Ces propriétés sont essentiellement:
1) la nature des cercles et des sphères (ou des variétés sphériques de dimension supérieure à deux);
2) les birapports de points (ou de variétés sphériques appartenant à un même faisceau, c'est à dire ayant en commun une variété sphérique dont la dimension est inférieure d'une unité).
Puisque l'angle de deux variétés sphériques s'exprime au moyen de birapports, la conservation des angles est une conséquence de la seconde propriété. De façon plus générale, on appelle transformation anallagmatique toute transformation jouissant des deux propriétés précédentes. L'ensemble de ces transformations est un groupe multiplicatif qui comprend les inversions, et les similitudes de E_m. Il a été étudié par Hadamard [15] dans le cas où m=3.

2. L'application K

Nous introduisons ici une application qui sera essentielle pour la suite de ce papier puisque, d'une part, elle nous conduira à la définition géométrique d'une étape de l'ε-algorithme tandis que, d'autre part, elle nous permettra de présenter les règles singulières de l'ε-algorithme. Cette application se définit à partir d'une application élémentaire H qui traduit directement la division harmonique de quatre points, l'un d'entre eux étant considéré comme une fonction des trois autres. La propriété la plus importante de ces applications est leur continuité. Pour ne pas alourdir inutilement l'exposé, nous passerons sous silence la plupart des propriétés géométriques attachées à l'application K; les plus élémentaires d'entre elles ont été présentées en [7].

2.1 La fonction H.

2.1.1 Définition.

Définissons formellement le fonction H de $(E_m)^3$ dans E_m par:

y=H(x,u,v) est le conjugué harmonique de x par rapport à u et v.

D'après la proposition 1, y=H(x,u,v) est défini si et seulement si les trois points x,u,v ne sont pas confondus. Nous notons $E_m^{(3)}$ l'ensemble des triplets $(x,y,z)\in(E_m)^3$ dont au moins un composant est distinct des deux autres, c'est à dire le complémentaire dans $(E_m)^3$ de l'ensemble $\{(x,y,z)\in(E_m)^3 | x=y=z\}$, lui-même isomorphe à E_m.

La fonction H est donc définie dans $E_m^{(3)}$ par: $y = H(x,u,v) = \begin{cases}(u+v)/2 & \text{si } x=\infty;\\ x+2[(u-x)^{-1}+(v-x)^{-1}]^{-1} & \text{si } x\neq\infty.\end{cases}$

On peut noter que cette dernière expression s'écrit encore: $y = H(x,u,v) = I_x([I_x(u)+I_x(v)]/2)$.

2.1.2 Continuité.

Proposition 2. *La fonction H est une application continue de* $E_m^{(3)}$ *dans* E_m.

Preuve: a) Montrons tout d'abord que, si le triplet $X=(x,u,v) \in E_m^{(3)}$ est tel que les quatre points x,u,v et $y=H(x,u,v)$ soient distincts de ∞ (c'est à dire si x,u, et v sont distincts de ∞ et $2x \neq u+v$), alors la fonction H est continue en X.

En effet, soit un second triplet $X'=(x',u',v')$ vérifiant cette condition:
$x',u',v' \neq \infty$ et $2x' \neq u'+v'$.
Un calcul élémentaire montre que, si nous posons: $\delta x=x'-x$, $\delta u=u'-u$ et $\delta v=v'-v$, nous obtenons:

$$H(x',u',v')-H(x,u,v) =$$
$$2\,\|u'+v'-2x'\|^{-2} * \{\, \|v'-x'\|^2\delta u + \|u'-x'\|^2\delta v - \|u'-v'\|^2\delta x/2 + \langle \delta v-\delta x | v-x+v'-x' \rangle (u-x) +$$
$$\langle \delta u-\delta x | u-x+u'-x' \rangle (v-x) + \langle \delta u+\delta v-2\delta x | u+v-2x+u'+v'-2x' \rangle \, [(u-x)^{-1}+(v-x)^{-1}]^{-1} \,\}$$

Puisque $u+v \neq 2x$, nous avons: $\rho = \|u+v-2x\| > 0$.
Pourvu que: $\max(\|\delta x\|,\|\delta u\|,\|\delta v\|) \leq \eta < \rho/4$,
nous pouvons écrire: $\|H(x',u,'v')-H(x,u,v)\| \leq F(x,u,v,\eta)$
avec:

$$F(x,u,v,\eta) = 2\eta/(\rho-4\eta) * \{\, \|v-x\| + \|u-x\| + \|u-v\|/2 + 5\eta +$$
$$4[(\|v-x\|+\eta)\|u-x\| \pm (\|u-x\|+\eta)\|v-x\|] + 8(\|u+v-2x\|+2\eta)\|[(u-x)^{-1}+(v-x)^{-1}]^{-1}\| \,\}.$$

Pour x,u et v fixés, cette majoration $F(x,u,v,\eta)$ est une fration rationnelle en η dont $\eta=0$ est racine et $\eta=\rho/4$ le pôle unique. Elle est donc continue en $\eta=0$:
Pour tout $\varepsilon>0$, il existe $\tau(\varepsilon)>0$ tel que $\eta<\tau(\varepsilon) \Longrightarrow F(x,u,v,\eta)<\varepsilon$.
Alors, pour tout $\varepsilon>0$, pour tout triplet $X'=(x,u,v)$ vérifiant:
$\max(\|x'-x\|,\|u'-u\|,\|v'-v\|) < \tau(\varepsilon)$,
nous avons:
$\|H(x',u',v')-H(x,u,v)\|<\varepsilon$,
ce qui assure la continuité de H en un triplet X de $E_m^{(3)}$ vérifiant les conditions indiquées.

b) Supposons maintenant que l'une des quatre conditions $x \neq \infty$, $u \neq \infty$, $v \neq \infty$ ou $y=H(x,u,v) \neq \infty$ ne soit pas réalisée.

Soit $t \in E_m$ distinct de x,u,v et y, ce qui implique: $t \in \mathbb{R}^m$.
Grace à l'invariance du birapport de quatre points dans une inversion, nous aurons:
$$I_t(H(x,u,v))=H(I_t(x),I_t(u),I_t(v)).$$
Posons alors: $\bar{x}=I_t(x)$, $\bar{u}=I_t(u)$, $\bar{v}=I_t(v)$ et $\bar{y}=H(\bar{x},\bar{u},\bar{v})$.
Pour établir la continuité de H en (x,u,v), il suffit de noter que pour t fixé distinct de x,u,v,y, nous avons:
1) I_t est continue sur E_m;

2) H est continue en $\bar{X}=(\bar{x},\bar{u},\bar{v})$ d'après (a);
en effet: $\bar{x},\bar{u},\bar{v}$ et $\bar{y}$ sont distincts de ∞ car s,u,v et y sont distincts de ∞
et la condition $(x,u,v) \in E_m^{(3)}$ assure $(\bar{x},\bar{u},\bar{v}) \in E_m^{(3)}$.

3) I_t est continue en $\bar{y}$.
Donc $H(x,u,v)=I_t(H(I_t(x),I_t(u),I_t(v)))$ est continue en (x,u,v).

Remarque: On peut donner de ce résultat une démonstration qui évite de particulariser le point à l'infini. Il suffit d'écrire H sous la forme: $H(x,u,v)=\varphi(\hat{H}(\varphi(x),\varphi(u),\varphi(v))$, où φ est l'isomor-

phisme topologique de E_m sur S_m et $\hat{H}$ la fonction qui, au triplet $(x,u,v) \in (S_m)^3 \setminus U_m$ (où $U_m=\{(r,s,t)\in(S_m)^3 | r=s=t\}$), associe le conjugué y de x par rapport à u et v dans S_m. En utilisant la même technique que dans la partie (a), on peut établir la continuité de $\hat{H}$ dans $S_m^{(3)}$ (défini de façon semblable à $E_m^{(3)}$). La continuité de φ (immédiate en raison de l'isomorphisme des topologies) permet alors de conclure.

2.2 La fonction K.

2.2.1 Définition.

Définissons formellement l'application K de $(E_m)^4$ dans E_m par:

$z = K(x,y,u,v) = H(y,x,h(x,u,v))$, soit: $z = H(y,x,w)$ avec $w = H(x,u,v)$.

L'application K est définie au point $(x,y,u,v)\in(E_m)^4$ si et seulement si les deux conditions suivantes sont remplies:

1) $w=H(x,u,v)$ est défini, ce qui est réalisé si et seulement si $(x,u,v) \in E_m^{(3)}$;

2) $z=H(y,x,w)$ est défini, ce qui est réalisé si et seulement si $(y,x,w) \in E_m^{(3)}$.

Or w n'est confondu avec x que dans le seul cas où l'un des points u ou v est confondu avec x. La fonction K est donc définie sur:

$$E_m^{(4)} = (E_m)^4 \setminus \{ (x,y,u,v) \in (E_m)^4 \mid x=y=u \text{ ou } x=u=v \text{ ou } x=v=y \}.$$

2.2.2 Continuité et expression de K.

L'application K est continue sur $E_m^{(4)}$ comme composée de deux fonctions continues. Puisque $w=H(x,u,v)$ et $z=H(y,x,w)$ traduisent respectivement $(w,x;u,v)=-1$ et $(z,y;x,w)=-1$, nous avons donc:

$$w=\begin{cases} I_x([I_x(u)+I_x(v)]/2) = I_x([I_x(y)+I_x(z)]/2) & \text{si } x\neq\infty \\ (u+v)/2=(y+z)/2 & \text{si } x=\infty. \end{cases}$$

soit:

$$w=\begin{cases} I_x(u)+I_x(v)=I_x(y)+I_x(z)=2I_x(w) & \text{si } x\neq\infty \\ u+v=y+z=2w & \text{si } x=\infty. \end{cases}$$

Si $x\neq\infty$, nous avons donc: $(u-x)^{-1}+(v-x)^{-1}=(y-x)^{-1}+(z-x)^{-1}=2(w-x)^{-1}$.

D'où l'expression de $z = K(x,y,u,v)$:

$$z=\begin{cases} x+[(u-x)^{-1}+(v-x)^{-1}-(y-x)^{-1}]^{-1} & \text{si } x\neq\infty \\ u+v-y & \text{si } x=\infty. \end{cases}$$

3. Vers une définition géométrique de l'ε-algorithme vectoriel

3.1 Forme losange de l'ε-algorithme.

L'ε-algorithme vectoriel a été initialement défini par Wynn [20] de la façon suivante:
$(x_n)_{n\geq 0}$ étant une suite de vecteurs de $\mathbb{R}^m$, on pose:

(3.1) $\quad \varepsilon_{-1}^{(n)}=0, \forall n\geq 1$ et $\varepsilon_0^{(n)}=x_n, \forall n \geq 0.$

et on utilise la relation:

(3.2) $\quad \varepsilon_{k+1}^{(n)}=\varepsilon_{k-1}^{(n+1)}+(\varepsilon_k^{(n+1)}-\varepsilon_k^{(n)})^{-1}, \forall k,n \geq 0.$

Les éléments $\varepsilon_k^{(n)}$ ainsi définis sont placés dans un tableau à double entrée où l'indice inférieur k repère une colonne et l'indice supérieur n une diagonale descendante. La relation (3.2) lie ainsi quatre points qui sont placés aux sommets d'un losange. Pour cette raison, nous nous refèrerons à cette définition comme à la forme losange de l'ε-algorithme. Le tableau sera construit systématiquement colonne par colonne vers la droite et de haut en bas en utilisant la relation (3.2).

3.2 Forme en croix de l' ε-algorithme.

Dans le tableau construit ci-dessus, les seules colonnes utiles sont celles dont l'indice est pair, les autres représentant des calculs intermédiaires. Wynn [22] a montré qu' on pouvait définir un algorithme ne faisant que les seules colonnes d'indice pair. On initialise:

(3.3) $\varepsilon_{-2}^{(n)}=\infty$, $\forall n\geq 2$ et $\varepsilon_0^{(n)}=x_n$, $\forall n\geq 0$,

et on utilise la relation:

(3.4) $(\varepsilon_{2k+2}^{(n)}-\varepsilon_{2k}^{(n+1)})^{-1}+(\varepsilon_{2k-2}^{(n+2)}-\varepsilon_{2k}^{(n+1)})^{-1}=(\varepsilon_{2k}^{(n+2)}-\varepsilon_{2k}^{(n+1)})^{-1}+(\varepsilon_{2k}^{(n)}-\varepsilon_{2k}^{(n+1)})^{-1}$, $\forall k,n \geq 0$.

Le tableau obtenu sera le tableau précédent restreint à ses seules colonnes d'indice pair. La relation (3.4) lie les valeurs aux quatre sommets d'une croix à la valeur au nœud de cette croix. Nous nous réfèrerons à cette définition comme à la forme en croix de l'ε-algorithme

3.3 L'ε-algorithme complexe.

Wynn a également défini l'ε-algorithme complexe [20], aussi bien avec la relation du losange qu'avec celle de la croix. Pour ce faire, il utilise l'inverse de Samelson d'un vecteur complexe (1.6) dans les relations (3.2) ou (3.4). En fait, il est indifférent de retenir comme inverse d'un vecteur l'inverse de Samuelson ou son conjugué ainsi que le montre la proposition suivante qui s'établit trivialement par récurrence sur l'indice de colonne k:

Proposition 3.1 *Soit $(x_n)_{n\geq 0}$ une suite de $\mathbb{C}^m$ et soient:*

1) $\varepsilon_k^{(n)}$ le tableau que fournit l'application de la forme losange de l'ε-algorithme en utilisant l'inverse de Samelson noté y^{-1}.

2) $\hat{\varepsilon}_k^{(n)}$ le tableau que fournit l'application cette même forme losange mais avec le conjugué de l'inverse de Samelson qu'on notera $y'=\bar{y}^{-1}$.

Alors on a: $\varepsilon_{2k}^{(n)}=\hat{\varepsilon}_{2k}^{(n)}$ *et* $\bar{\varepsilon}_{2k-1}^{(n)}=\hat{\varepsilon}_{2k-1}^{(n)}$, $\forall k,n \geq 0$.

Remarque: Ainsi que nous l'avons remarqué en (1.1.3), ceci entraîne que l'isomorphisme ψ de G_m sur E_{2m} respecte les colonnes paires du tableau construit à partir d'une suite de $\mathbb{C}^m$ donnée. Plus précisément, si l'on note $\varepsilon_{2k}^{(n)}(x_i)$ les éléments du tableau construit à partir d'une suite $(x_i)_{i\geq 0}$ donnée, nous avons: $\psi(\varepsilon_{2k}^{(n)}(x_i)) = \varepsilon_{2k}^{(n)}(\psi(x_i))$.

3.4 Définition géométrique de l' ε-algorithme vectoriel.

Compte tenu de l'expression analytique de l'application k, la relation (3.4) est équivalente à:

$\varepsilon_{2k}^{(n)} = K(\varepsilon_{2k}^{(n+1)}, \varepsilon_{2k-2}^{(n+2)}, \varepsilon_{2k}^{(n)}, \varepsilon_{2k}^{(n+2)})$, où $\varepsilon_{2k}^{(n+1)} \neq \infty$.

Il est donc possible de donner une définition équivalente à celle de (3.3) et (3.4) par:

$\varepsilon_{-2}^{(n)}=\infty$, $\forall n\geq 2$, et $\varepsilon_0^{(n)}=x_n$, $\forall n\geq 0$;

$\varepsilon_{2k}^{(n)}= K(\varepsilon_{2k}^{(n+1)}, \varepsilon_{2k-2}^{(n+2)}, \varepsilon_{2k}^{(n)}, \varepsilon_{2k}^{(n+2)})$, pour tout couple (k,n) tel que:

$$\begin{cases} 1^\circ\ \varepsilon_{2k}^{(n+1)}\neq\infty \\ 2^\circ \text{ au plus un des trois points } \varepsilon_{2k-2}^{(n+2)} \text{ ou } \varepsilon_{2k}^{(n)} \text{ ou } \varepsilon_{2k}^{(n+2)} \text{ est confondu avec } \varepsilon_{2k}^{(n+1)}. \end{cases}$$

Pourvu que cette dernière condition soit satisfaite, la fonction K reste continue au voisinage de $\varepsilon_{2k}^{(n+1)}=\infty$. On peut alors prolonger par continuité la définition de l'algorithme en éliminant cette condition et donner de l'ε-algorithme vectoriel la définition géométrique suivante:

Initialisation: $\varepsilon_{-2}^{(n)}=\infty$, $\forall n\geq 2$, et $\varepsilon_0^{(n)}=x_n$, $\forall n\geq 0$;

Pour tout couple d'indices (k,n) satisfaisant:

$$\begin{cases} 1^\circ\ k\geq 0 \text{ et } n\geq 0 \\ 2^\circ \text{ au plus } \begin{cases} \text{un des trois points } \varepsilon_{2k-2}^{(n+2)} \text{ ou } \varepsilon_{2k}^{(n)} \\ \text{ou } \varepsilon_{2k}^{(n+2)} \text{ est confondu avec } \varepsilon_{2k}^{(n+1)}; \end{cases} \end{cases}$$

effectuer les deux opérations suivantes:

$$\begin{cases} 1^\circ \text{déterminer } \gamma_{2k}^{(n+1)} \text{: conjugué de } \varepsilon_{2k}^{(n+1)} \text{ par rapport à } \varepsilon_{2k}^{(n)} \text{ et } \varepsilon_{2k}^{(n+2)}; \\ 2^\circ \text{déterminer } \varepsilon_{2k+2}^{(n)} \text{: conjugué de } \varepsilon_{2k-2}^{(n+2)} \text{ par rapport à } \varepsilon_{2k}^{(n+1)} \text{ et } \gamma_{2k}^{(n+1)}. \end{cases}$$

La traduction analytique de cet algorithme consiste à substituer à la règle de la croix (3.4) la règle suivante quand $\varepsilon_{2k}^{(n+1)}=\infty$:

$$\varepsilon_{2k+2}^{(n)} = \varepsilon_{2k}^{(n)}+\varepsilon_{2k}^{(n+2)}-\varepsilon_{2k-2}^{(n+2)}.$$

On retrouve la règle singulière introduite par Wynn [21] dans le cas de l'ε-algorithme scalaire et étendue par l'auteur à l'ε-algorithme vectoriel [8].

3.5 Intérêt et limites de la définition géométrique.

Le premier intérêt de cette interprétation géométrique est d'étendre la règle singulière de Wynn à l'ε-algorithme vectoriel. Toutefois, l'interprétation géométrique n'est pas essentielle puisque le prolongement par continuité nous conduit à ce résultat sans qu'il soit nécessaire d'interpréter géométriquement l'application K. Par contre, cette interprétation permet de donner une définition qui recouvre la règle de la croix classique et la règle singulière. Ces deux règles apparaissent alors comme la traduction analytique d'une fonction d'itération définie géométriquement au moyen de la seule notion de division harmonique. Puisque cette notion est respectée par les transformations anallagmatiques, nous avons le résultat suivant:

Proposition 3.2 *Si nous notons $\varepsilon_{2k}^{(n)}(x_i,\alpha)$ les éléments du tableau construit à partir de l'initialisation:* $\varepsilon_{-2}^{(n)}=\alpha\in E_m$ $\forall n\geq 2$, et $\varepsilon_0^{(n)}=x_n\neq\alpha$, $\forall n\geq 0$;

Alors pour toute transformation anallagmatique A, pour tout couple d'indices non négatifs (n,k), nous avons:

$$\varepsilon_{2k}^{(n)}(A(x_i),A(\alpha)) = A(\varepsilon_{2k}^{(n)}(x_i,\alpha)).$$

L'interprétation géométrique ne permet pas de définir l'algorithme dans le cas où plusieurs valeurs consécutives d'une même colonne sont confondues. Toutefois, en l'appliquant aux colonnes d' indice impair, on pourrait étendre la règle singulière au cas où deux valeurs consécutives sont égales. En fait, cette extension n'est qu'un cas particulier de celle que nous allons maintenant présenter.

4.Quadrillages

Dans ce chapitre, nous définirons deux sortes de quadrillages qui serviront de base pour l' introduction des ε-tableaux associés à la mise en œuvre de l'ε-algorithme. Bien que très élémentaires, ces notions ne semblent pas avoir été étudiées jusqu'à présent.

4.1 losanges et croix.

Décomposons l'ensemble de tous les couples de **Z**x**Z** en deux parties complémentaires disjointes:

$P = \{(x,y)\in \mathbb{Z}^2 \mid x+y\equiv 0 \text{ (modulo 2)}\}$

$I = \{(x,y)\in \mathbb{Z}^2 \mid x+y\equiv 1 \text{ (modulo 2)}\}$

Chacune de ces parties se décompose encore en deux parties complémentaires disjointes dans chacune desquelles une **composante** fixée conserve la même parité (ce qui implique la même propriété pour l'autre composante). Nous noterons:

$P=P_0\cup P_1$, où $P_i=\{(x,y)\in P \mid x\equiv i \pmod 2\}$ pour $i=0,1$.

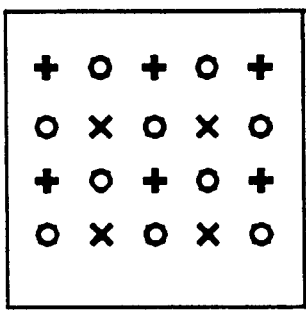


figure 1

partition de $\mathbb{Z}\times\mathbb{Z}$

o éléments de I

\+ éléments de P_0

× éléments de P_1

On appellera ***losange de centre*** *$X=(x,y)\in I$, l'ensemble des quatre points:*

$L(X)=\{(x+1,y),\ (x-1,y),\ (x,y+1),\ (x,y-1)\}$.

Ces quatre points, qui appartiennent à P, sont les ***sommets*** *du losange. Deux sommets sont dits* ***opposés*** *si leur milieu est le centre du losange,* ***contigus*** *dans le cas contraire.*

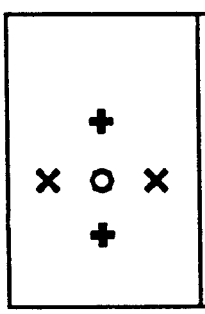


figure 2

éléments d'un losange

o centre du losange

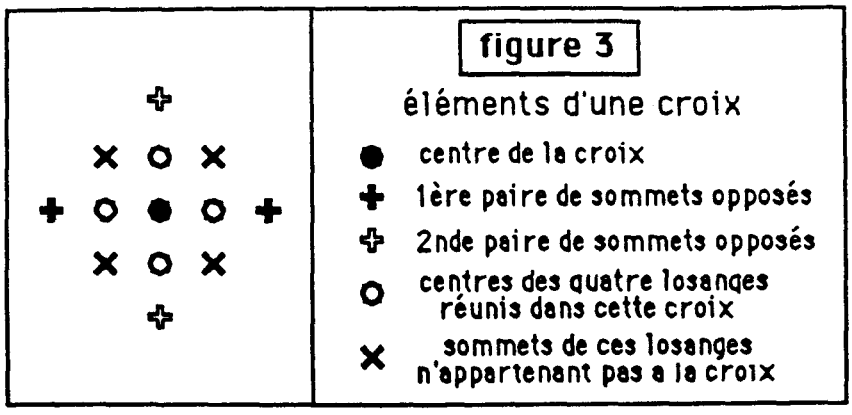
\+ 1ère paire de sommets opposés

× 2nde paire de sommets opposés

On appellera ***croix*** *de centre $X=(x,y)\in P$ l'ensemble des cinq points:*

$C(X)=\{(x,y),\ (x+2,y),\ (x-2,y),\ (x,y-2),\ (x,y+2)\}$.

Les quatre derniers points sont les ***sommets*** *de la croix.* Ils appartiennent à la même composante P_i que celle à laquelle appartient le centre de la croix. Comme pour le losange, *deux sommets sont* ***opposés*** *si leur milieu est le centre de la croix, et* ***contigus*** *dans le cas contraire.*



Nous dirons que deux points X et Y sont **voisins** *si la croix centrée en X compte Y parmi ses sommets: deux points voisins appartiennent à la même composante* P_i.

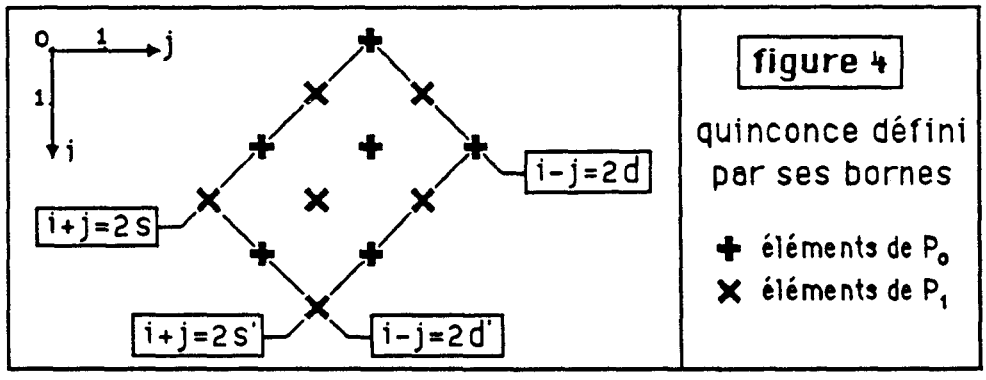
4.2 Quadrillages en quinconce.

4.2.1 Quinconce et base de quinconce.

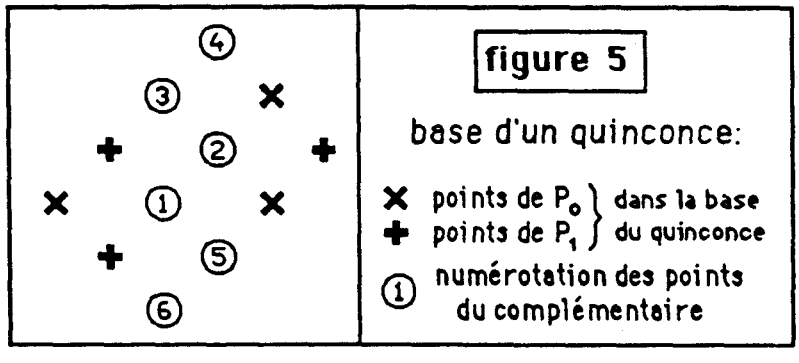
Etant donnés quatre entiers relatifs s, s', d et d' vérifiant s≤s' et d≤d', on appellera quadrillage en quinconce ou plus brièvement **quinconce** *de* **bornes** *s,s', d et d' (prises dans cet ordre) l'ensemble Q défini par:*

$$Q = \{(x,y) \in P \mid 2s \le x+y \le 2s',\ 2d \le x-y \le 2d'\}$$

Le quinconce de bornes (s,s',d,d') sera dit **diagonal** *si (s'-s)(d'-d)=0.*
Il est diagonal **montant** *si s=s', diagonal* **descendant** *si d=d'.*



Une partie B d'un quinconce Q est une **base** *de Q si les deux conditions suivantes sont remplies:*
1° B ne contient aucun losange.
2° Il existe une numérotation des points X_i $(i=1,\dots,n)$ *du complémentaire de B dans Q (noté Q\B) telle que, si l'on pose:* $\Omega_0=B$ *et* $\Omega_i=\Omega_{i-1}\cup\{X_i\}$ $(i=1,\dots,n)$, *alors* X_i *appartient à un seul losange de* Ω_i $(i=1,\dots,n)$.



Remarque: Cela revient à dire qu'il existe une bijection $X_i \leftrightarrow L_i$ (i=1,...,n) de l'ensemble des points $X_i \in Q\backslash B$ sur l'ensemble L(Q) des losanges du quinconce et que cette bijection vérifie:

$X_i \in L_i$ et $X_j \notin L_i$, $\forall i<j$.

4.2.2 Propriétés élémentaires.

Proposition 4.1. *Toutes les bases d'un quinconce comptent le même nombre de points.*

Preuve: Par définition, une base est une partie B du quinconce Q telle qu'il existe une bijection de l'ensemble Q\B sur l'ensemble des losanges de Q. Ces deux ensembles ont même cardinal: le nombre de points de la base est donc égal à la différence entre le nombre de points du quinconce et le nombre de losanges contenus dans ce quinconce: il est indépendant du choix de la base.

Corollaire. *Toute base d'un quinconce de bornes s,s',d et d' compte s'-s+d'-d+1 points.*

Preuve: Il suffit de dénombrer le nombre n_p de points et le nombre n_L de losanges du quinconce Q. On vérifie immédiatement que $n_p = (s'-s+1)(d'-d+1)$ et $n_L = (s'-s)(d'-d)$. D'où le nombre de points de la base: $n_B = n_p - n_L$.

Proposition 4.2. *Le quinconce Q de bornes s,s',d et d' est la réunion des d'-d+1 quinconces diagonaux descendants: $D_\delta = \{(x,y) \in P / 2s \leq x+y \leq 2s',\ x-y=2\delta\}$, $\delta=d,\ldots,d'$. Chacun de ces quinconces diagonaux D_δ contient au moins un point de toute base B de Q.*

Preuve: Le premier point est immédiat. Notons que tous les D_δ comptent le même nombre de points, ce qui implique $D_\delta \neq \emptyset$. Etablissons le second point en raisonnant par l'absurde: supposons qu'il existe une base B et un indice δ compris entre d et d' tel que la base B ne contienne aucun point de D_δ. Puisque B est une base, il existe une suite Ω_i (i=0,...,n) de parties de Q vérifiant:

$$\begin{cases} \Omega_0 = B \\ \Omega_i = \Omega_{i-1} \cup \{X_i\} \quad (i=1,\ldots,n) \\ \Omega_n = Q \\ X_i \text{ appartient à un et un seul losange de } \Omega_i. \end{cases}$$

Notons i le plus petit indice tel que Ω_i contienne un point Y de D_δ. Un tel indice existe puisque $\Omega_n = Q$ contient D_δ non vide, et il est strictement positif car $\Omega_0 = B$. Par définition, ce point Y appartient à Ω_i sans appartenir à Ω_{i-1}: c'est donc le point X_i et il appartient à un losange de Ω_i. C'est même le seul point de D_δ qui appartienne à ce losange. Ceci revient à dire qu'il existe un losange de Q dont l'intersection avec la diagonale D_δ est réduite à un point. Ceci est impossible car l' intersection d'un losange avec une diagonale ne peut compter que 0 ou deux points.

Proposition 4.3. *Le quinconce Q de bornes s,s',d et d' est la réunion des quinconces diagonaux montants $D'_\sigma = \{(x,y) \in P / 2d \leq x-y \leq 2d',\ x+y=2\sigma\}$, $\sigma=s,\ldots,s'$. Chacun de ces quinconces diagonaux D'_σ contient au moins un point de toute base B de Q.*

La **démonstration** est identique à celle de la proposition précédente.

Proposition 4.4. *Si B_1 et B_2 sont deux bases respectives des deux quinconces Q_1 et Q_2 et si B_1 est inclus dans B_2 alors Q_1 est inclus dans Q_2*

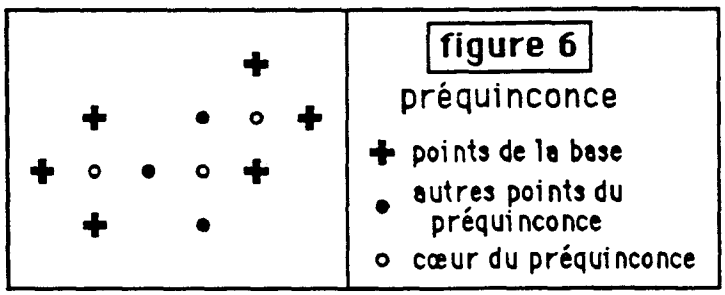
Démonstration: Soient s_i, s'_i, d_i et d'_i les bornes de Q_i ($i=1,2$). Puisque B_1 est une base de Q_1, à tout σ tel que $s_1 \leq \sigma \leq s'_1$, on peut associer un point $(x,y) \in B_1$ tel que $x+y=2\sigma$ (d'après la proposition 4.2). Puisque B_1 est inclus dans B_2 lui-même inclus dans Q_2, on a: $2s \leq x+y \leq 2s'$ soit $2s \leq 2\sigma \leq 2s'$, et ceci pour tout $\sigma \in \{s_1,\ldots,s'_1\}$. D'où $s_2 \leq s_1 \leq s'_1 \leq s'_2$. En utilisant la proposition 4.3 on montrera de même que $d_2 \leq d_1 \leq d'_1 \leq d'_2$. D'où l'inclusion de Q_1 dans Q_2.

Corollaire. *Si B est une base des deux quiconces Q_1 et Q_2, ces deux quinconces sont confondus.*

Preuve: D'après la proposition 4.4,l'inclusion de la base B de Q_1 dans la base B de Q_2 nous assure celle de Q_1 dans Q_2. Par symétrie, Q_2 est aussi inclus dans Q_1 d'où l'identité des deux quinconces.

4.3 Préquinconces.

On appellera **préquiconce** *P d'un quinconce Q toute partie P de Q contenant une base de Q. On appellera cœur d'un préquiconce P la partie de I constituée par l'ensemble des centres des losanges contenus dans ce préquinconce.* Par exemple, les ensembles Ω_i introduits lors de la définition des bases de quinconce sont des préquinconces dont le cœur comporte i éléments. En particulier, une base est un préquinconce dont le cœur est vide.



Remarque: Une base donnée ne pouvant être base que d'un seul quinconce (corollaire de la proposition 4.4), il y a unicité du quinconce utilisé dans la définition d'un préquinconce. Si P est un préquinconce de Q, toute base de Q contenue dans P sera appelée **base** du préquinconce P. (Par définition, il en existe au moins une).

4.4 Bases simples.

La notion de base de quinconce introduite en 4.2.1 est d'un usage trop délicat. Nous introduisons ici une notion plus restreinte, mais de manipulation sera plus aisée:

Définition: *Une base d'un quinconce Q est dite* **simple** *s'il est possible de numéroter les points b_i ($i=1,\ldots,p$) de cette base de façon que, pour $i=1,\ldots,p$, l'ensemble $B_i = \bigcup_{j=1}^{i} (b_j)$ soit une base d'un quinconce Q_i.* En général, une base n'est pas simple: c'est le cas de la base de la figure 6.

Proposition 4.5. (illustrée par la figure 7) *Soit $B=(b_j, j=1,\ldots,p)$ une base simple d'un quinconce Q et soient $B_i = \bigcup_{j=1}^{i} (b_j)$, les bases des quinconces intermédiaires Q_i, $i=1,\ldots,p$. Il existe une suite de quinconces diagonaux $(Q_i)_{i=1\ldots p}$ tels que :*

$$\begin{cases} Q_1 = B_1 = (b_1) \\ b_i \in q_i \text{ et } Q_i = Q_{i-1} \cup q_i \text{ (pour } i=1\ldots p) \end{cases}$$

Preuve: Notons s_i,s'_i,d_i,d'_i les bornes du quinconce Q_i. D'après le corollaire de la proposition 4.1, sa base comptera $s'_i-s_i+d'_i-d_i+1$ points. Puisque d'après la proposition 4.4, Q_{i-1} est inclus dans Q_i, on a: $s_i \leq s_{i-1} \leq s'_{i-1} \leq s'_i$ et $d_i \leq d_{i-1} \leq d'_{i-1} \leq d'_i$, et la base B_{i-1} comptera $s'_{i-1}-s_{i-1}+d'_{i-1}-d_{i-1}+1$ points. Alors $B_i=B_{i-1}\cup\{b_i\}$ implique $(s'_i-s_i+d'_i-d_i+1)-(s'_{i-1}-s_{i-1}+d'_{i-1}-d_{i-1}+1)=1$ soit:

$$(s'_i-s'_{i-1})+(d'_i-d'_{i-1})+(d_{i-1}-d_i)+(s_{i-1}-s_i)=1.$$

Chacune des quatre différences parenthésées étant entière et non négative, on en conclut que trois de ces différences sont nulles, la quatrième valant 1: ceci correspond à l'adjonction d'un quinconce diagonal que nous notons q_i. Il reste à montrer que $b_i \in q_i$. A cette fin, notons que $b_j \in Q_{i-1}$ implique $b_j \notin q_i$, pour $j=1,\ldots,i-1$. Si b_i n'appartenait pas à q_i, alors le quinconce diagonal q_i ne comporterait aucun point de la base B_i de Q_i, ce qui contredit l'une des deux propositions 4.2 ou 4.3.

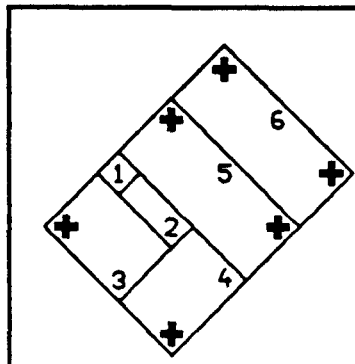


figure 7

base simple de quinconce

on a exhibé une numérotation convenable (non unique) des points d'une base simple de ce quinconce, et on a encadré les six quinconces correspondant à cette numérotation.

4.3 Quadrillages en grille.

4.3.1 Définitions: grille et base de grille.

Soient $i \in \{0,1\}$ et quatre entiers relatifs s,s',d,d' vérifiant $s \leq s'$ et $d \leq d'$.

*On appelle quadrillage en grille (ou plus brièvement **grille**) de P_i de **bornes** s,s',d,d' (prises dans cet ordre) l'ensemble G défini par:* $G_i = \{(x,y) \in P_i \mid 2s \leq x+y \leq 2s',\ 2d \leq x-y \leq 2d'\}$.

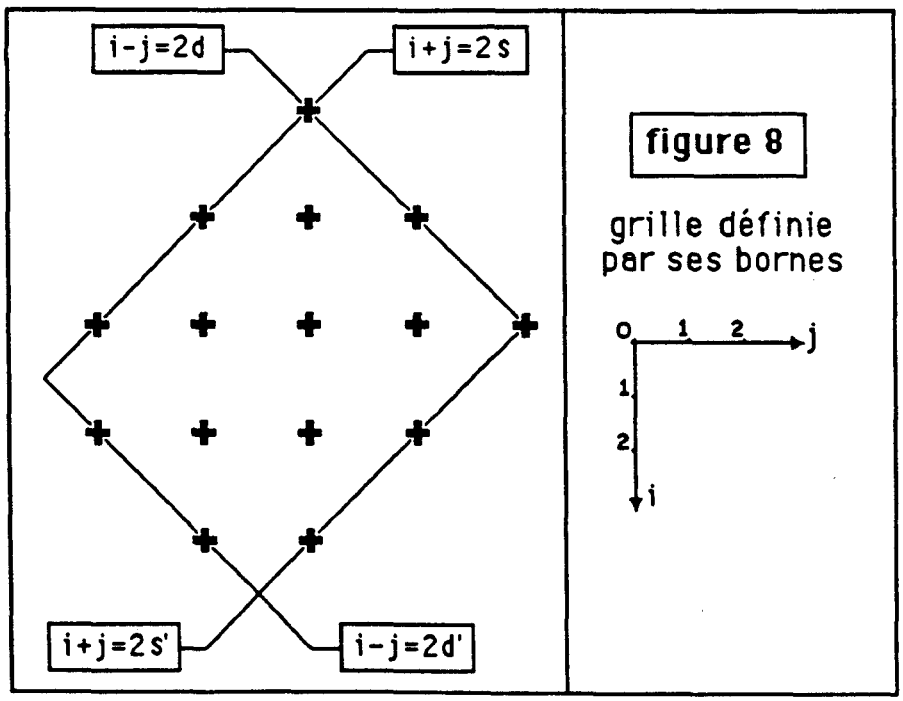
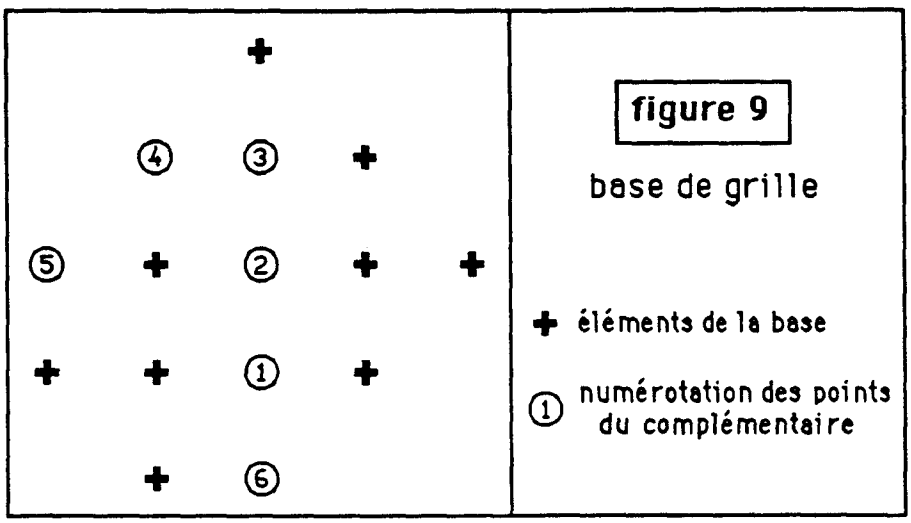


figure 8

grille définie par ses bornes

C'est aussi l'intersection de P_i avec le quinconce de même bornes: un quinconce Q induit sur P_i une grille G_i de mêmes bornes, pour i=0,1 et on a $Q=G_0 \cup G_1$.

Nous dirons qu'une grille de P_i est ***diagonale****, diagonale* ***montante*** *ou diagnale* ***descendante*** *si c'est l'intersection avec P_i d'un quinconce de cette nature.*
Une partie B d'une grille G est une ***base*** *de G si les deux conditions suivantes sont remplies:*
1° B ne contient aucune croix.
2° Il existe une numérotation des points X_i (i=1,...,n) du complémentaire G\B telle que, si l'on pose: $\Omega_0=B$ et $\Omega_i=\Omega_{i-1}\cup\{X_i\}$ (i=1,...,n), alors X_i appartient à une seule croix de Ω_i et en est un sommet (i=1,...,n).
Remarque: Cela revient à dire qu'il existe une bijection $X_i \leftrightarrow C_i$ (i=1,...,n) de l'ensemble des points $X_j \in G\backslash B$ sur l'ensemble C(G) des croix de la grille et que cette bijection vérifie: $X_i \in C_i$ et $X_j \notin C_i$, $\forall i<j$.



4.3.2 Propriétés élémentaires.

Par des démonstrations très voisines de celles qui ont été utilisées pour les quinconces (et que, par conséquent, nous ne détaillerons pas), on établit les résultats suivants:

Proposition 4.6. *Toutes les bases de grille comptent le même nombre de points.*

Proposition 4.7. *La grille G de P_i de bornes s,s',d,d' est la réunion des d'-d+1 grilles diagonales descendantes: $D_\delta=\{(x,y)\in P_i / 2s\le x+y\le 2s',\ x-y=2\delta\}$, $\delta=d,...,d'$. Chacune de ces grilles D_δ si elle est non vide, contient au moins un point de toute base B de G. La grille G est aussi la réunion des grilles diagonales montantes $D'_\sigma=\{(x,y)\in P_i / 2d\le x-y\le 2d',\ x+y=2\sigma\}$, $\sigma=s,...,s'$. Chacune de ces grilles D'_σ si elle est non vide, contient au moins un point de toute base B de G.*

Remarque: Notant en passant que les grilles diagonales D_δ (ou D'_δ) ne comptent pas nécessairement le même nombre de points. Par exemple, si la grille g est diagonale montante, ce sera la réunion de grilles diagonales descendantes dont la moitié seront vides, les autres comportant un seul élément: la restriction "si elle est non vide" n'est pas gratuite.

Proposition 4.8. *Si B_1 et B_2 sont deux bases respectives des deux grilles G_1 et G_2 et si B_1 est inclus dans B_2 alors G_1 est inclus dans G_2*

Corollaire. *Si B est une base commune aux deux grilles G_1 et G_2,, alors ces deux grilles sont confondues.*

3.3 Dénombrement des éléments d'une grille.

Alors que le dénombrement des éléments d'un quinconce était particulièrement simple (corollaire de la proposition 4.1), celui des éléments d'une grille est compliqué par la nécessité de tenir compte de la parité de la composante P_i à laquelle appartient le grille. Cette question est résolue par les quatre propositions qui suivent:

Proposition 4.9. *Les centres des croix de la grille G de P_i de bornes s,s',d,d' sont les points de la grille G_1 de P_i de bornes s+1,s'-1,d+1,d'-1.*

Preuve: Une croix de centre (x,y) est contenue dans G si et seulement si le point (x,y) et ses quatre voisins (x±2,y), (x,y±2) appartiennent à G. Cette condition s'exprime par les vingt inégalités suivantes:

$2s\leq x+y\leq 2s'$, $2s\leq (x+2)+y\leq 2s'$, $2s\leq (x-2)+y\leq 2s'$, $2s\leq x+(y+2)\leq 2s'$, $2s'\leq x+(y-2)\leq 2s'$,
$2d\leq x-y\leq 2d'$, $2d\leq (x+2)-y\leq 2d'$, $2d\leq (x-2)-y\leq 2d'$, $2d\leq x-(y+2)\leq 2d'$, $2d'\leq x-(y-2)\leq 2d'$.

On vérifie immédiatement l'équivalence de ce système avec:
$2s+2\leq x+y\leq 2s'-2$ et $2d+2<x-y\leq 2d'-2$,
c'est à dire le système qui caractérise l'appartenance du centre (x,y) à G_1.

Corollaire. *Le nombre de croix de la grille G de P_i est égal au nombre de points de la grille G_1 de P_i de bornes s+1,s'-1,d+1,d'-1.*

Proposition 4.10. *Si la grille G de P_i (i=0 ou 1) de bornes s, s', d, d' est telle que l'un des deux écarts s'-s ou d'-d soit impair, alors elle compte (s'-s+1)(d'-d+1)/2 points.*

Démonstration: Etablissons ce résultat dans le cas où la différence s'-s est impaire.

A tout point (x,y) de G, on associe le point (x',y') défini par x'=s'+s-y et y'=s'+s-x. Ce point vérifie: x'-y'=x-y et (x'+y')+(x+y)=2(s+s'). Alors la condition $2s\leq 2(s+s')\leq 2s'$ est équivalente à: $2s\leq 2(s+s')-(x'+y')\leq 2s'$, soit à: $2s\leq x'+y'\leq 2d'$, tandis que $2d\leq x-y\leq 2d'$ équivaut à $2d\leq x'-y'\leq 2d'$. Ainsi, l'application F:(x,y)→(x',y') applique la grille de P_i dans le quinconce Q de mêmes bornes.
Puisque G est inclus dans P_i, on a x-y=i (mod 2). D'autre part:
s'-s=1 (mod2) entraîne x'=1-y (mod2) et y'=1-x (mod2), d'où x'=y'=1-i (mod2),
et alors $(x',y')\in P_{1-i}$. Donc F applique la grille G de P_i dans la grille G' de P_{1-i} de mêmes bornes.
En prenant i'=1-i, on voit que F applique la grille G' de P_{1-i} dans la grille G de P_i: F est donc une bijection de G sur G': ces deux grilles ont même cardinal. Or le quinconce Q de mêmes bornes vérifie Q=G∪G' et compte (s'-s+1)(d'-d+1) points. D'où le résultat dans ce cas.
Dans le cas où la différence d'-d est impaire, un raisonnement analogue conduit au même résultat.

Proposition 4.11. *Si la grille G de P_i de bornes s,s',d,d' est telle que les deux différences s'-s et d'-d soient paires, alors la grille G compte:*
(s'-s+1)(d'-d+1)+1]/2 points si s+d et i ont la même parité;
(s'-s+1)(d'-d+1)-1]/2 points sinon.

Démonstration: La figure 10 permet de suivre plus commodément cette démonstration.
La translation T: (x,y)→(x+1,y+1) transforme la grille G en une grille G_1 de P_{1-i} de bornes s+1,s'+1,d+1,d'+1. Introduisons les deux grilles diagonales:

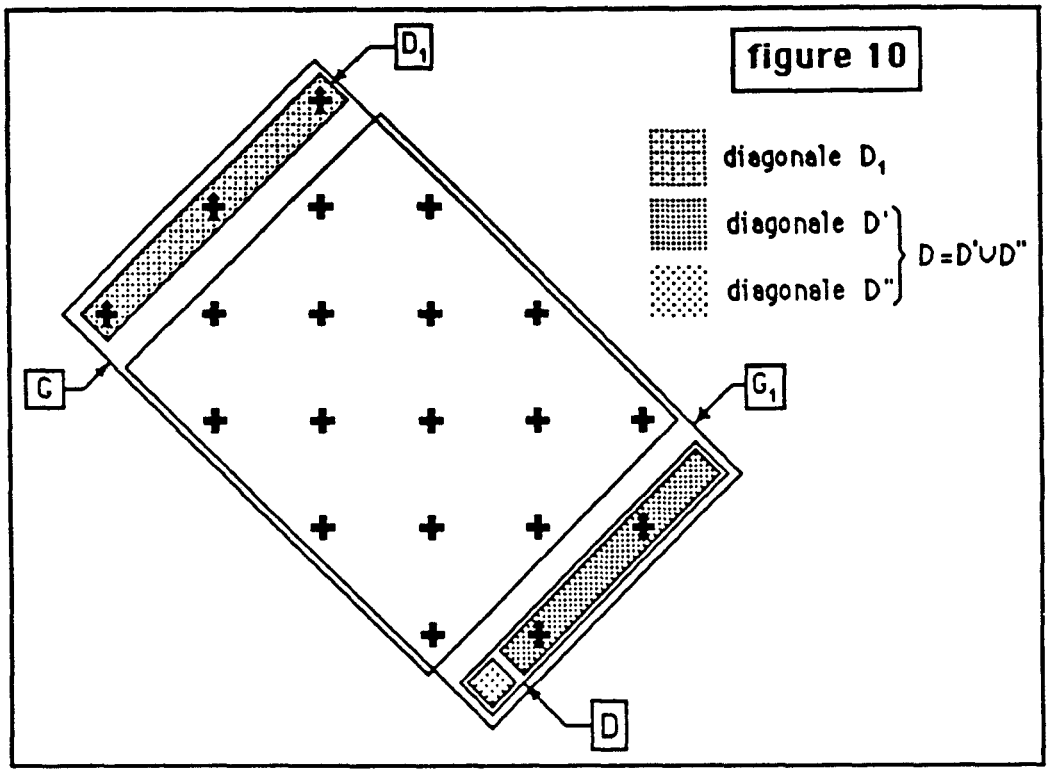
$$\begin{cases} D=\{(x,y)\in P_i \mid 2d\leq x-y\leq 2d' \text{ et } x+y=2(s'+1)\} \\ D_1=\{(x,y)\in P_{1-i} \mid 2d\leq x-y\leq 2d' \text{ et } x+y=2s\}. \end{cases}$$

Dénombrons les éléments de D: D s'écrit sous la forme D=D'∪D", où $D'=\{(x,y)\in P_i \mid 2d\leq x-y\leq 2d'-2, x+y=2s'+2\}$ et $D''=\{(x,y)\in P_i \mid x-y=2d' \text{ et } x+y=2s'+2\}$. Puisque d'-d est pair, on peut appliquer la proposition 4.10 à D': D' compte (d'-d)/2 points.

D'autre part, le quinconce qui a les bornes de D" compte un seul point, le point (x,y)=(d'+s'+1,s'-d'+1). Ce point vérifie donc x=y=s+d+1 (mod 2), et il n'appartient à D" que si s+d+1=i (mod 2), c'est à dire si s+d et i sont de parité différente.

D comptera donc: $\begin{cases}(d'-d)/2 \text{ points} & \text{si } s+d \text{ et } i \text{ ont la même parité}\\ (d'-d)/2+1 \text{ points} & \text{sinon.}\end{cases}$

Les deux ensembles G∪D et $G_1 \cup D_1$ sont respectivement les grilles de P_i et P_{1-i} sur les mêmes bornes s,s'+1,d,d'. D'après la proposition 4.10, G∪D compte donc (s'-s+2)(d'-d+1)/2 points. Il suffit de retrancher les points de D pour aboutir au résultat annoncé.



Proposition 4.12. *Si la grille G de P_i de bornes s,s',d,d' n'est pas reduite à une grille diagonale, alors toute base de G comptera s'-s+d'-d points.*

démonstration: D'après les deux propositions précédentes, la grille G compte:

$$[(s'-s+1)(d'-d+1)+\alpha_i]/2 \text{ points}$$

$$\text{où } \alpha_i=\begin{cases}0 & \text{si } (s'-s+1)(d'-d+1) = 0 \pmod 2\\ 1 & \text{si } (s'-s+1)(d'-d+1) = 1 \pmod 2 \text{ et } d+s-i=0 \pmod 2\\ -1 & \text{si } (s'-s+1)(d'-d+1) = 1 \pmod 2 \text{ et } d+s+i=1 \pmod 2\end{cases}$$

Si s'-s≥2 et d'-d≥2, alors la grille de P_i de bornes s+1, s'-1, d+1, d'-1 (dont les points sont en bijection avec les croix de G) comporte $[(s'-s+1)(d'-d+1)+\alpha_i]/2$ points. D'où, par différence, le nombre de points de la base.

Si l'une des différences s'-s ou d'-d est égale à 1, la grille ne comporte aucune croix: elle est donc identique à sa base. Il suffit alors de dénombrer le nombre de points de la grille en notant que nous sommes alors dans les conditions d'application de la proposition 4.10:

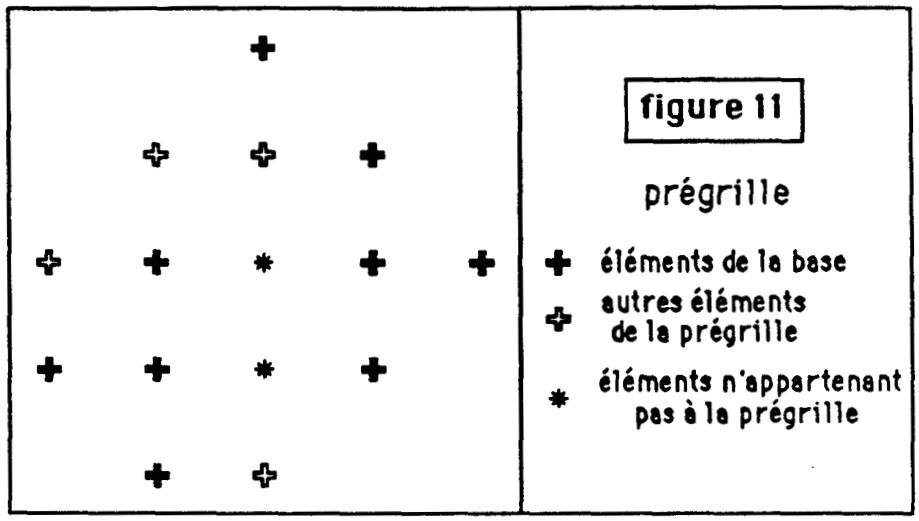
si s'-s=1, on a: $n_B=(d'-d+1)(s'-s+1)=d'-d+1=d'+d+s'-s$;

si d'-d=1, on a: $n_B=(d'-d+1)(s'-s+1)=s'-s+1=s'-s+d'-d$.

4.3.4 Prégrilles.

*On appellera **prégrille** P de la grille G toute partie P de G contenant une base de G. On appelle **cœur** d'une prégrille P la partie de P constituée par les centres des croix contenues dans P.*

Par exemple, les ensembles Ω_1 introduits lors de la définition des bases de grille sont des prégrilles et les bases de grilles sont des prégrilles dont le cœur est vide. Comme pour les quinconces, une prégrille ne peut être prégrille que d'une seule grille. Si P est une prégrille de G, toute base de G incluse dans P sera appelée base de la prégrille.

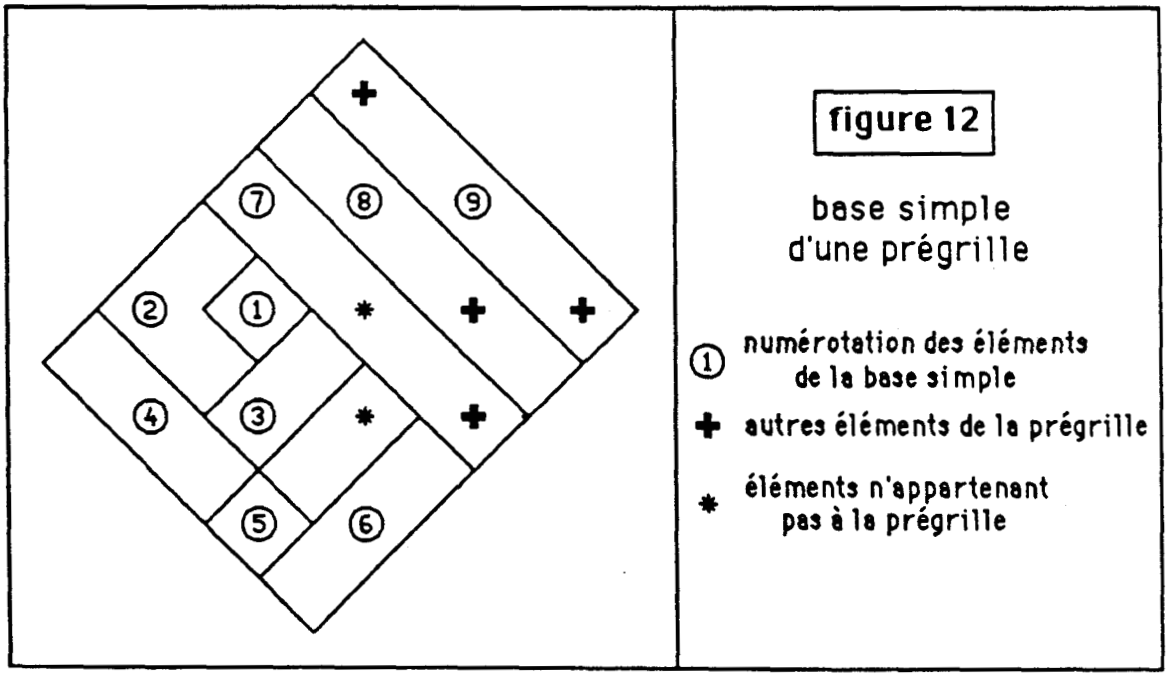


4.3.5 Bases simples de grille.

Une base B d'une grille G est dite **simple** *s'il est possible de numéroter les points* b_k $(k=1,\ldots,p)$ *de cette base de façon que, pour* $k=1,\ldots,p$, *l'ensemble* $B_k = \bigcup_{j=1}^{k} (b_j)$ *soit une base d'une grille* G_k.

Toute prégrille définie au moyen d'une base simple est dite simple.

En général, une base de grille n'est pas simple: c'est le cas de la base introduite dans la figure 9.



Proposition 4.13. *Soit* $B=(b_j, j=1,\ldots,p)$ *une base simple d'une grille G de* P_i, *à laquelle est*

associée la suite finie $B_i=\bigcup_{j=1}^{i}(b_j)$ *)) des bases des grilles intermédiaires* G_i, $i=1,\dots,p$. *Alors, il existe une suite de grilles diagonales* $(g_i)_{i=1\dots p}$ *tels que :*

$$\begin{cases} G_1=B_1=(b_1) \\ b_i\in g_i \text{ et } G_i=G_{i-1}\cup g_i \text{ (pour } i=2\dots p) \end{cases}$$

Preuve: Notons s_i, s'_i, d_i, d'_i les bornes de la grille G_i. Supposons que $(s'_i-s_i)(d'_i-d_i)>0$. D'après la proposition 4.12, sa base comptera $s'_i-s_i+d'_i-d_i$ points. D'après la proposition 4.8, G_{i-1} est incluse dans G_i, on a: $s_i\le s_{i-1}\le s'_{i-1}\le s'_i$ et $d_i\le d_{i-1}\le d'_{i-1}\le d'_i$, et la base B_{i-1} compte $s'_{i-1}-s_{i-1}+d'_{i-1}-d_{i-1}+1$ points. Alors $B_i=B_{i-1}\cup(b_i)$ implique: $(s'_i-s_i+d'_i-d_i)-(s'_{i-1}-s_{i-1}+d'_{i-1}-d_{i-1})=1$, soit:

$$(s'_i-s'_{i-1})+(d'_i-d'_{i-1})+(d_{i-1}-d_i)+(s_{i-1}-s_i)=1.$$

Un raisonnement semblable à celui mis en œuvre pour les quinconces dans la proposition 4.5 conduit à l'existence d'une grille diagonale g_i satisfaisant $b_i\in g_i$ et $G_i=G_{i-1}\cup g_i$ dès que: $(s'_i-s_i)(d'_i-d_i)>0$. Dans le cas où la grille G_i est diagonale, un raisonnement élémentaire mais assez laborieux montre que $G_i=G_{i-1}\cup g_i$ où g_i est réduit à un seul élément qui n'est autre que b_i.

4.4 Sur quelques quadrillages particuliers.

4.4.1 Sur les bases.

Dans les paragraphes 6 et 7, nous serons amenés à considérer des quadrillages en grille ou en quiconce d'une structure assez particulière pour lesquels il nous sera très utile de pouvoir exhiber une base simple. les deux propositions qui suivent répondent à cette attente.

Pour ne pas alourdir exagérément les notations, nous poserons:
$\Lambda_i^j(n)=\{(i,l)\in\mathbb{Z}^2 \mid l=j+2k,\ k=0,\dots,n-1\}$ et $\Gamma_i^j(n)=\{(l,j)\in\mathbb{Z}^2 \mid l=i+2k,\ k=0,\dots,n-1\}$.

Nous avons alors les deux résultats suivants:

Proposition 4.14. *Soient trois entiers n,p,q vérifiant* $n\ge 1$ *et* $0\le p,q\le n$.
Alors l'ensemble $B_i^j(n,p,q)=\Gamma_{i+2}^j(n)\cup\Gamma_{i+2}^{j+2}(n)\cup\Lambda_i^{j+2}(p)\cup\Lambda_{i+2n+2}^{j+2}(q)$ *est une base simple de la grille G de* P_λ *de bornes:*$i+j+2, i+j+2n+2q+2, i-j-2p, i-j+2n$, *où* $\lambda\in\{0,1\}$ *désigne la parité de* $i+j$ *(c'est à dire:* $i+j=\lambda\ (mod 2)$*).*

Proposition 1.15. *Sous les mêmes conditions, l'ensemble* $\tilde{B}_i^j(n,p,q)=B_i^j(n,p,q)\cup\{(i+1,j+1)\}$ *est une base simple du quinconce Q de mêmes bornes.*

Nous ne donnerons pas ici la démonstration de ces deux propriétés. Dans chaque cas, il suffit d'exhiber deux numérotations convenables:
1° une numérotation des points du complémentaire de la base telle que l'addition de chaque nouveau point augmente le nombre de losanges (ou de croix) d'une unité, ce qui montre qu'on a bien une base;
2° une numérotation des points de la base telle que chaque sous-base intermédiaire soit base d'un sous-quinconce (ou d'une sous-grille), ce qui montre que la base est simple.

Il s'agit d'un exercice simple mais fastidieux dont nous esquissons ci-dessous sous forme de figures une solution dans un cas particulier (n=3,p=2,q=1).

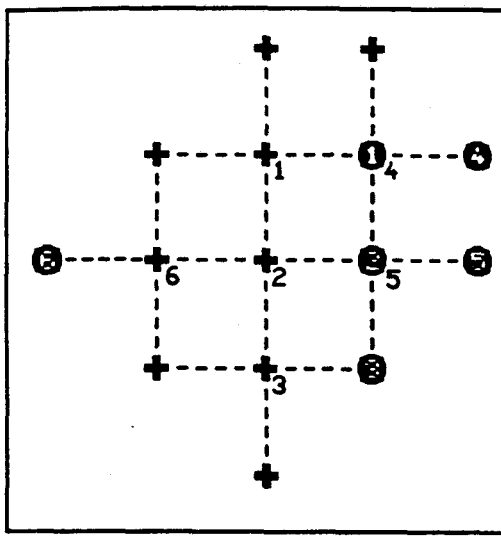


figure 13

base d'une grille particulière

points de la base

numérotation du complément de la base dans la grille

numérotation des centres des croix associées au complément de la base

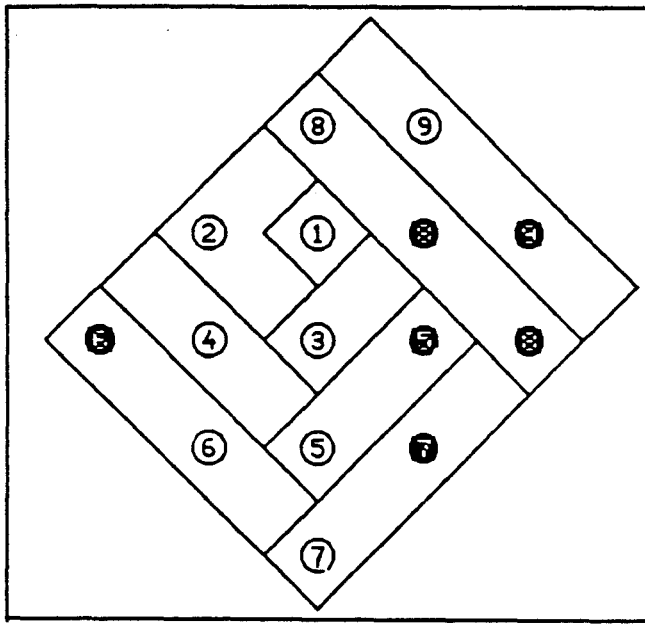


figure 14

base simple d'une grille

① numérotation des points de la base

autres éléments de la grille

on a encadré chacune des 9 grilles G_i dont les bases respectives B_i sont:

$$B_i = \bigcup_{j=1}^{i} \{b_j\}$$

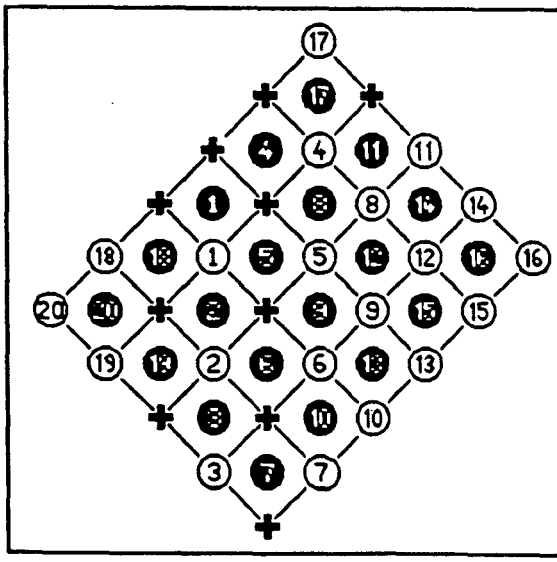


figure 15

base d'un quinconce

points de la base

① numérotation des points du complément de la base

numérotation des centres des losanges associés

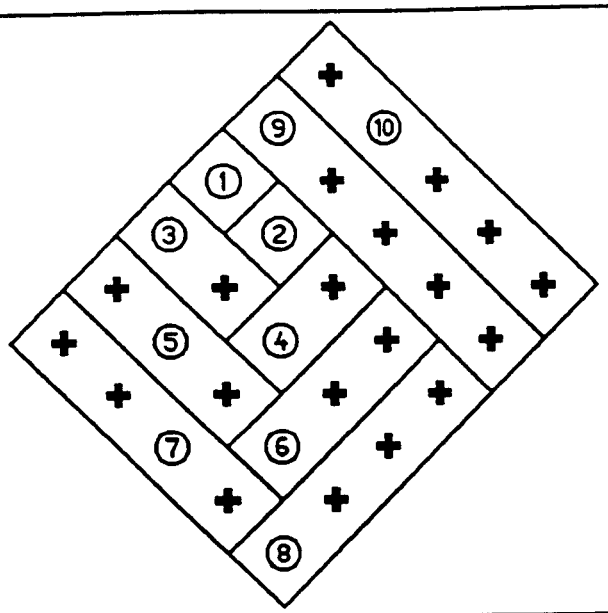


figure 16

base simple d'un quinconce

① numérotation des points de la base simple retenue

on notera que cette numérotation déduit de celle de la figure 14 par un simple décalage d'indice.

✚ complément de la base sans le quinconce

on a encadré les quinconces successifs O_i de base B_i définie par: $B_i = \bigcup_{j=1}^{i} \{b_j\}$

4.4.2 Sur les carrés.

Dans les paragraphes qui suivent, il sera commode de désigner sous forme concise l'ensemble des points d'une grille qui sont localisés à l'intérieur d'un carré. A cette fin, nous introduirons encore les définitions suivantes qui sont illustrées par la figure 17:

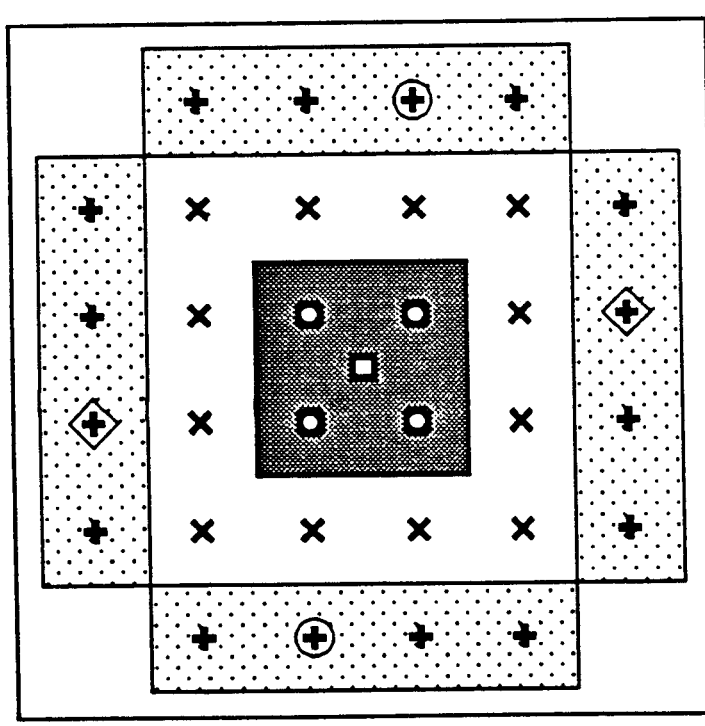


figure 17

carré de coté 4

ceinture du carré

frontière du carré

intérieur du carré

centre du carré

1er couple de points opposés

2nd couple de points opposés

les deux couples de points opposés choisis sont symétriques l'un de l'autre.

Soient l,l',c,c' des entiers relatifs vérifiant: $\gamma = l'-l+1 = c'-c+1 \geq 1$ et $i \in \{0,1\}$.

On appelle ***carré*** *de* P_i *de* ***bornes*** $2l+i, 2l'+i, 2c+i, 2c'+i$ *et de coté* γ *l'ensemble:*

$$C = \{(x,y) \in P_i \mid 2l+i \leq x \leq 2l'+i,\ 2c+i \leq y \leq 2c'+i\}.$$

*Dans un tel carré, on distingue l'****intérieur****:* $\overset{\bullet}{C} = \{(x,y) \in P_i \mid 2l+i < x < 2l'+i,\ 2c+i < y < 2c'+i\}$,

et la ***frontière****:* $Fr(C) = C \setminus \overset{\bullet}{C}$. *En outre on appelle* ***ceinture*** *du carré C la réunion des croix centrées sur C et privée de C.*

En particulier, l'ensemble ne comportant qu'un point d'une grille est un carré de côté 1, d'intérieur vide, et dont la ceinture compte quatre points.

Deux points (j,k) et (j',k') appartenant à la ceinture d'un carré C de P_i de bornes 2l+i, 2l'+i, 2c+i et 2c'+i seront dits ***opposés*** *par rapport à ce carré si l'on a:*

$$j+j'=2(l+l'+i) \quad et \quad k+k'=2(l+l'+i).$$

Deux paires de points opposés par rapport au carré C: ((j,k) et (j',k'), d'une part, (j_1,k_1) et (j'_1,k'_1) d'autre part) seront dits ***symétriques*** *par rapport à C si l'on a:*

$$|j-j'|=|k_1-k'_1| \quad et \quad |k-k'|=|j_1-j'_1|.$$

5. Les ε-tableaux

On utilise les notions de quadrillage pour définir les notions d'ε-tableaux en croix et en losange et on caractérise ceux qui sont normaux. Le mode de représentation retenu permet de munir l'ensemble des ε-tableaux d'une structure topologique. Les deux résultats essentiels sont la densité des ε-tableaux normaux dans l'ensemble de tous les ε-tableaux et la liaison entre la normalité des ε-tableaux en losange et celle des ε-tableaux en croix qui s'en déduisent par restriction.

5.1 Les ε-tableaux en losange.

5.1.1 définitions.

Soit Q un quinconce, P un préquinconce de Q, de cœur C_p et A une partie non vide de E_m.

On appelle ***ε-tableau en losange*** *(en abrégé: ε-TL) défini sur P à valeurs dans A une application a de P dans A, qu'on notera $(a_{(i,j)})_{(i,j)\in P}$ ou plus brièvement a_p et qui vérifie:*

$$(a_{(i+1,j)}-a_{(i-1,j)})^{-1}=a_{(i,j+1)}-a_{(i,j-1)}, \ \forall (i,j)\in C_p$$

Remarque. Cette définition implique que les expressions qui figurent dans les deux membres soient définies, c'est à dire que l'on n'ait ni $a_{(i+1,j)}=a_{(i-1,j)}=\infty$, ni $a_{(i,j+1)}=a_{(i,j-1)}=\infty$. Par contre, un des quatre termes peut être infini: ceci entraîne alors que les deux termes qui figurent dans l'autre membre soient égaux, mais distincts de ∞. Dans la pratique, on se limite aux choix $A=E_m$ ou $A=\mathbb{R}^m$.

Soient P_1 inclus dans P_2 des préquinconces respectifs des quinconces Q_1 et Q_2.

Tout ε-TL a_{P_2} (défini sur P_2) induit sur P_1 un ε-TL qu'on appellera ***restriction*** *de a_{P_2} à P_1. L'opération inverse de la restriction est une opération multivoque dont chaque occurrence sera appelée* ***extension*** *de a_{P_1} à P_2.* Une telle extension peut ne pas exister. *Un ε-TL a_p sur le préquinconce P sera dit* ***normal*** *s'il est la restriction à P d'un ε-tableau de $\mathbb{R}^m$ sur un quinconce contenant P.* Cette notion de normalité d'un ε-tableau est étroitement liée à celle d'une table de Padé [11].

5.1.2 Topologie sur les ε-TL.

L'ensemble A_p des applications d'un préquinconce P dans $(E_m)^{Card(P)}$ est un espace topologique dès qu'on le munit de la topologie de l'espace produit $\prod_{(i,j)\in P} E_m$. *Notons L_p le sous-ensemble de A_p constitué par les ε-TL sur P dans E_m.* La topologie de A_p induit sur L_p une structure d'espace topologique. L'espace A_p est compact en tant que produit d'espaces compacts. Il s'ensuit que L_p sera relativement compact. Plus précisément, on a le résultat suivant:

Proposition 5.1. *Pour que L_p soit compact, il suffit que P soit une base de quinconce.*

Démonstration: Si P est est une base, son cœur est vide. Il y a donc identité de L_p et A_p et la compacité de A_p assure le résultat.

5.1.3 Densité des ε-TL normaux.

Proposition 5.2. *Soit B une base simple du quinconce Q. L'ensemble des ε-TL normaux sur B est dense dans L_B*

Demonstration: D'après la définition d'une base simple, il est possible de numéroter les n+1 éléments b_i de B de façon que l'ensemble $B_i = \bigcup_{j=1}^{i} \{b_j\}$ soit une base du quinconce Q_i pour $i=0,\dots,n$, avec: $Q_0=B_0$, $Q_n=Q$ et Q_{i-1} inclus dans Q_i pour $i=1,\dots,n$. D'après la proposition 4.5, Q_i peut s'écrire: $Q_i = Q_{i-1} \cup q_i$, où q_i est un quinconce diagonal contenant b_i, $i=1,\dots,n$.
Fixons l'indice i: $1 \le i \le n$. Montrons tout d'abord que, si a est un ε-TL sur Q_{i-1} dans $\mathbb{R}^m$ (c'est à dire normal), il existe un et un seul ε-TL a' sur Q_i dans E^m tel que:

$a'_l = a_l \ \forall l \in Q_{i-1}$ et $a'_{b_i} = x$ où x est un élément donné de E^m.

Puisque q_i est un quinconce diagonal contenant b_i, il existe une suite (finie) de quinconces diagonaux $(q_i^k)_{k=0\dots k_i}$, vérifiant l'initialisation $q_i^0 = \{b_i\}$ et la récurrence $q_i^k = q^{k-1} \cup \{\alpha_k\}$, $k=1,k_i$, où les a_i sont les éléments de q_i pris dans ordre convenable. Chacun des ensembles $Q_i^k = Q_{i-1} \cup q_i^k$ contient un point α_k et un losange L_k de plus que le précédent pour $k=1,\dots,k_i$. Ceci permet de déterminer de façon unique les valeurs a'_{a_k} en utilisant la relation du losange. En effet, chacun des losanges L_k comporte deux sommets contigus appartenants à Q_i, donc distincts de ∞. La valeur du troisième sommet inclus dans Q_i^{k-1} définit donc celle du quatrième sommet (même si la valeur au troisième sommet est confondue avec ∞, comme on l'a vu en remarque 1).

Montrons maintenant qu'il existe un nombre fini de valeurs $x \in E_m$ telles que la condition $a'_{b_i} = x$ implique que a' n'est pas un ε-TL normal sur Q_i. Pour qu'il en soit ainsi, il faut et il suffit qu'en un des points a_K de la diagonale q_i, on ait $a'_{\alpha_k} = \infty$. Notons que, pour k fixé, l'ensemble $B_{i-1} \cup \{\alpha_k\}$ est une base de Q_i. En appliquant le résultat précédent à cette base, on voit que $a'_{\alpha_k} = \infty$ implique que a_{b_i} prenne une valeur x_k parfaitement définie, et ceci pour $k=0,\dots,k_i$.
Il s'ensuit que l'ensemble des ε-TL normaux sur B_i est dense dans l'ensemble des ε-TL définis sur B_i dont la restriction à B_{i-1} est normale: en effet, parmi les ε-TL définis sur B_i dont la restriction à B_{i-1} est normale, seuls un nombre fini k_i+1 d'entre eux ne sont pas normaux.

On peut encore traduire ce dernier résultat par:
L'ensemble des ε-TL définis sur B dont la restriction à B_i est normale est dense dans l'ensemble des ε-TL définis sur B dont la restriction à B_{i-1} est normale, et ceci, pour $i=1,\dots,n$.

La transitivité de la densité assure alors le résultat annoncé.

5.2 Les ε-tableaux en croix.

5.2.1 définitions.

Soit G une grille, P une prégrille de G, de cœur C_p et A une partie non vide de E_m.

On appelle **ε-tableau en croix** *(en abrégé: ε-TC) défini sur P à valeurs dans A une application a de P dans A, qu'on notera* $(a_{(i,j)})_{(i,j)\in P}$ *ou plus brièvement* a_p *et qui vérifie:*

$$a_{(i,j+2)}=K(a_{(i,j)},a_{(i,j-2)},a_{(i-2,j)},a_{(i+2,j)}),\ \forall (i,j)\in C_p$$

Soient P_1 inclus dans P_2 deux prégrilles respectives des grilles G_1 et G_2.

Tout ε-TC a_{P_2} *défini sur* P_2 *induit sur* P_1 *un ε-TC qu'on appellera* **restriction** *de* a_{P_2} *à* P_1.

L'opération inverse de la restriction est une opération multivoque dont chaque occurrence sera appelée **extension** *de* a_{P_1} *à* P_2. Une telle extension peut ne pas exister.

Un ε-TC a_p *sur la prégrille P sera dit* **normal** *s'il est la restriction à P d'un ε-TC de* $\mathbb{R}^m$ *sur une grille G contenant P telle que, pour tout couple (u,v) de points voisins, on ait:* $a_u \neq a_v$

5.2.2 Topologie sur les ε-TC.

L'ensemble A_p des applications d'une prégrille P dans $(E_m)^{Card(P)}$ est un espace topologique compact dès qu'on le munit de la topologie de l'espace produit $\prod_{(i,j)\in P} E_m$. *Notons* C_p *le sous-ensemble de* A_p *constitué par les ε-TC sur P dans* E_m. La topologie de A_p induit sur C_p une structure d'espace topologique. L'espace A_p est compact en tant que produit d'espaces compacts. Il s'ensuit que C_p sera relativement compact. Dans le cas particulier où P est une base de grille, C_p est compact.

5.2.3 Densité des ε-TC normaux.

Proposition 5.3. *Soit B une base simple de la grille G. L'ensemble des ε-TC normaux sur B est dense dans* C_B

Démonstration: La démonstration se calque sur celle de la proposition 5.2 relative à la densité des ε-TL normaux. Nous ne la détaillerons pas ici.

5.3 Opérations sur les ε-tableaux.

5.3.1 Opérations sur le support.

Certaines opérations sur le support (préquinconce ou prégrille) d'un ε-tableau ne modifient pas la nature de cet ε-tableau. Ces opérations consistent essentiellement en une modification de la numérotation des points du support et, pourvu qu'elles préservent la position relative de ces points, elles n'ont aucune incidence sur les ε-tableaux définis sur ce support. Ces opérations sont:

la translation:	$(i,j)\rightarrow(i+k,j+l)$, $\forall k,l\in\mathbb{Z}$
la symétrie verticale:	$(i,j)\rightarrow(-i,j)$
la symétrie horizontale:	$(i,j)\rightarrow(i,-j)$
la transposition:	$(i,j)\rightarrow(j,i)$

et tout produit de ces opérations.

Cette invariance des propriétés d'un ε-tableau par rapport à la numérotation du support permettra d'établir des propriétés particularisant une orientation ou une direction.

5.3.2 Opérations sur les ε-TC.

L'opération K étant préservée dans toute transformation anallagmatique (produit d'isométries, homotéties ou inversions), le transformé anallagmatique point à point d'un ε-TC sera encore un ε-TC. Cette propriété permettra de traiter le cas où des points d'un ε-TC deviennent infinis en les banalisant au moyen d'une inversion.

5.4 Liaison des ε-TL et des ε-TC.

Bien qu'elles aient été introduites indépendamment l'une de l'autre, les notions d'ε-TL et d'ε-TC sont étroitement liées par le résultat suivant:

Proposition 5.4. *Soit Q un quinconce de bornes s,s',d,d' et soient $G_\mu = Q \cap P_\mu$ (μ=0,1) les deux grilles associées à ce quinconce.*

(i) Si t est un ε-TL sur Q, alors la restriction de t à G_μ est un ε-TC.

(ii) De plus, si t est normal alors sa restriction à G_μ est aussi normale.

(iii) Si t est un ε-TC normal sur G_μ et B_μ une base de G_μ, et si $B=B_\mu \cup (b)$ est une base simple de Q (avec $b \in G_{1-\mu}$), alors, à tout $a \in RM$ on peut associer un et un seul ε-TL u normal sur B tel que : $u_\lambda = t_\lambda$, $\forall \lambda \in G_\mu$ et $u_b = a$.

Démonstration. Etablissons successivement les trois points (i), (ii), (iii).

(i) Il suffit de montrer que, pour toute croix de G_μ, la relation de la croix est satisfaite:
Soient (i,j), (i+2,j), (i-2,j), (i,j-2) et (i,j+2) les éléments d'une croix de G_μ. Alors les quatre points (i+1,j+1), (i+1,j-1), (i-1,j+1) et (i-1,j-1) appartiennent à Q et on a les quatre relations:

$$t_{(i+1,j+1)} - t_{(i-1,j+1)} = (t_{(i+2,j)} - t_{(i,j)})^{-1}$$
$$t_{(i+1,j-1)} - t_{(i-1,j-1)} = (t_{(i,j)} - t_{(i,j-2)})^{-1}$$
$$t_{(i+1,j+1)} - t_{(i+1,j-1)} = (t_{(i+2,j)} - t_{(i,j)})^{-1}$$
$$t_{(i-1,j+1)} - t_{(i-1,j-1)} = (t_{(i,j)} - t_{(i-2,j)})^{-1}$$

En ajoutant membre à membre la première et la quatrième équation d'une part, la seconde et la troisième d'autre part, on obtient:

$$(t_{(i,j+2)} - t_{(i,j)})^{-1} - (t_{(i-2,j)} - t_{(i,j)})^{-1} = (t_{(i+2,j)} - t_{(i,j)})^{-1} - (t_{(i,j-2)} - t_{(i,j)})^{-1}$$

La règle de la croix est bien vérifiée.

(ii) Si l'ε-TL t est normal, alors $t_\lambda \neq \infty$, $\forall \lambda \in Q$, et a fortiori, $\forall \lambda \in G_\mu$. Il reste à montrer que, si λ et ν sont voisins dans G_μ, alors $t_\lambda \neq t_\nu$. En effet, λ et ν sont opposés dans le losange centré en $(\lambda+\nu)/2$. Notons λ' et ν' les deux autres sommets de ce losange. Si $t_\lambda = t_\nu$, on a: $t_{\lambda'} - t_{\nu'} = \pm(t_\lambda - t_\nu)^{-1} = \infty$, ce qui est impossible car $t_{\lambda'}$ et $t_{\nu'}$ sont distincts de ∞.

(iii) Puisque B est une base simple de Q, il est possible de numéroter ses n+1 éléments b_i de façon que $B_i = \bigcup_{j=1}^{i} (b_j)$ soit base d'un quinconce Q_i pour i=0,...,n. Il est aisé de voir que $b \in B_1 = Q_1$, ce qui assu-re la normalité sur Q_1 de l'ε-TL défini par:

$$u_l = \begin{cases} t_l & \text{si } l \in B_i \setminus \{b\} \\ \alpha & \text{si } l = b \end{cases}$$

Il existe donc $i \geq 2$ et un ε-TL u normal sur Q_{i-1} tel que: $u_l = \begin{cases} t_l & \text{si } l \in Q_{i-1} \cap P_\mu \\ \alpha & \text{si } l = b \end{cases}$

Imposons maintenant $u_{b_i} = t_{b_i}$. De la même façon qu'en 5.1.3, la normalité de u sur Q_{i-1} nous permet de construire une extension de u sur Q_i. La restriction à $G_\mu \cap Q_i$ de cette extension est un ε-TC identique à t sur la base $B_\mu \cap Q_i$ de $G_\mu \cap Q_i$. Elle est donc identique à t sur $G_\mu \cap Q_i$.

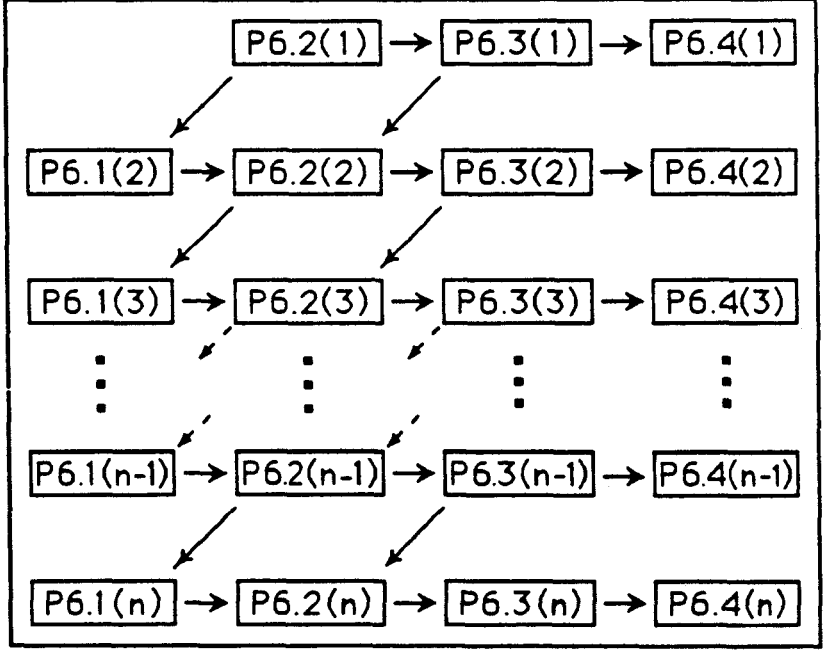
Supposons que cette extension ne soit pas normale sur Q_i. Il existe donc un indice $l \in q_i$ tel que $u_l = \infty$, ce qui équivaut encore à: il existe un losange de Q_i contenant l dont deux sommets opposés λ et v vérifient: $u_\lambda = u_v$. Aucun des trois points l, λ, ou v ne peut donc appartenir à $G_\mu \cap Q_i$. La contradiction provient du fait que trois sommets d'un losange n'appartiennent pas au même P_μ. L'extension est donc normale sur Q_i, $\forall i \leq n$.

6. Limites d'ε-TC normaux

6.1 Introduction.

Dans ce paragraphe, nous nous intéressons à des ε-TC définis sur des prégrilles particulières. Nous allons montrer que, si une suite d'ε-TC normaux est telle que sa restriction à une certaine base (simple) tende vers un ε-TC satisfaisant certaines conditions, alors cet ε-TC limite peut être défini sur une prégrille au moyen de nouvelles règles généralisant les règles classiques.

figure 18 articulation des démonstrations



Ce résultat sera énoncé au moyen de la proposition 6.4. Pour l'établir, nous ferons appel aux trois propositions 6.1, 6.2 et 6.3 dont la démonstration se fait par récurrence simultanée. En effet, chacune de ces propositions énonce une propriété relative à un entier positif n. Pour pouvoir démontrer la proposition 6.1 pour $n \geq 2$, il faut avoir établi la proposition 6.2 pour n-1, tandis que la preuve de la proposition 6.2 pour $n \geq 2$ fait appel à la proposition 6.3 pour n-1. D'autre part, pour n fixé, la proposition 6.2 se déduit de la proposition 6.1 et la proposition 6.3 de la proposition 6.2.

Le processus est initialisé par le fait que la proposition 6.2 est immédiate pour n=1. Enfin, pour tout n, la proposition 6.4 se déduit de la proposition 6.3. Ce schéma de démonstration est décrit par la figure 18.

6.2 Enoncé des trois propriétés.

Dans les trois propositions qui suivent, n est un entier strictement positif.

Proposition 6.1. *Soit t un ε-TC défini sur la base $B_i^j(n-1,n-1,1)$ tel qu'il existe $x\in R^m$ vérifiant:* $\forall\lambda \in \Gamma_{i+2}^{j+2}(n) = \Gamma_{i+2}^{j+2}(n-1) \cup \Lambda_{2n+2+i}^{j+2}(1) \Rightarrow t_\lambda = x$

et : $\forall\lambda \in \Gamma_{i+2}^{j}(n-1) \cup \Lambda_{i}^{j+2}(n-1) \Rightarrow t_\lambda \neq x$

Alors pour toute suite $u^{(k)}$ d'ε-TC normaux dont la restriction à $B_i^j(n-1,n-1,0)$ converge vers t, on a: $\lambda\in\Lambda_{i+2}^{j+2}(n) \Rightarrow \lim_{k\to\infty} u_\lambda^{(k)} = x$

Proposition 6.2. *Soit t un ε-TC défini sur la base $B_i^j(n,n,p)$ (avec $1\le p\le n$) et tel qu'il existe $x\in R^m$ vérifiant:* $\forall\lambda\in\Gamma_{i+2}^{j+2}(n) \Rightarrow t_\lambda = x$ *et* $\forall\lambda\in B_i^j(n,n,p)\setminus\Gamma_{i+2}^{j+2}(n)) \Rightarrow \lambda \neq x$.

Alors pour toute suite $u^{(k)}$ d'ε-TC normaux dont la restriction à $B_i^j(n,n,p)$ converge vers t, on a, en posant $p'=\min(p+1,n)$:

$$\forall\lambda\in\Lambda_{i+2}^{j+2}(n)\cup\Lambda_{i+2n}^{j+2}(p')\cup\Gamma_{i+2}^{j+2n}(p') \Rightarrow \lim_{k\to\infty} u_\lambda^{(k)} = x$$

$$et\ m\in(1,\dots,p') \Rightarrow \lim_{k\to\infty} u_{(i+2m,j+2n+2)}^{(k)} = K(x, t_{(i-2n+2-2m,j)}, t_{(i,j+2n+2-2m)}, t_{(i+2n+2,j+2m)})$$

Proposition 6.3. *La proposition 6.2 reste vraie si $x=\infty$.*

6.3 Démonstration de la proposition 6.1

Pour établir la validité de cette proposition pour une valeur $n\ge 2$ donnée, nous supposons démontrée la proposition 6.3 pour la valeur n-1. Pour n=1, une telle hypothèse est inutile. Il est clair que l'on peut se contenter d'établir la proposition pour i=j=0.

D'après la proposition 4.14, $B_0^0(n-1,n-1,1)$ est une base simple de la grille G_{n-1} de bornes 2,2n+2,-2n+2,2n-2 (voir figure 19). Puisque les ε-TC $u^{(k)}$ sont normaux sur $B_0^0(n-1,n-1,1)$, en tout point $\lambda\in\Lambda_2^2(n)$ contenu dans G_{n-1} on a $u_\lambda^{(k)}\in R^m$.

Comme E_m est compact, la suite $(u^{(k)})$ admet au moins une valeur d'accumulation. Si, pour tout $\lambda\in\Lambda_2^2(n)$, ces valeurs d'accumulation ne sont pas distinctes de x, le résultat est établi.

C'est en particulier le cas pour le point $\lambda=(2,4)$. En effet, les valeurs

$$u_{(2,4)}^{(k)} = u_{(2,2)}^{(k)} + [(u_{(4,2)}^{(k)} - u_{(2,2)}^{(k)})^{-1} + (u_{(0,2)}^{(k)} - u_{(2,2)}^{(k)})^{-1} - (u_{(2,0)}^{(k)} - u_{(2,2)}^{(k)})^{-1}]^{-1}$$

tendent vers: $\lim_{k\to\infty} u_{(2,4)}^{(k)} = x + [(x-x)^{-1} + (t_{(0,2)}-x)^{-1} - (t_{(2,0)}-x)^{-1}]^{-1} = x.$

La proposition est donc bien établie pour n=1.

Dans le cas contraire, notons m le plus petit indice tel que la suite $u^{(k)}_{(2,2m)}$ ait une valeur d'adhérence y distincte de x. Puisque $\lim_{k\to\infty} u^{(k)}_{(2,4)}=x$, cet indice m est strictement compris entre 1 et n.

Considérons alors le sous-ensemble de la grille G_{n-1}:

$$\bar{B}_{(0,0)}(m-1,m-1,1)= \Lambda_{(2,2)}(m) \cup \Lambda_{(2,0)}(m-1) \cup \Gamma_{(0,2)}(m-1).$$

C'est une base de la grille G'_{m-1} de bornes 2,2m+2,-2m+2,2m-2. On note $(u^{(r)})_{r\in \mathbb{N}'}$ la sous-suite de $(u^{(k)})_{k\in \mathbb{N}}$ telle que la suite de terme général $u^{(r)}_{2\ 2m}$ tende vers y quand r tend vers l'infini dans $\mathbb{N}'$. Cette suite d'ε-TC (normaux) est telle que sa restriction à $\bar{B}^0_0(m-1,m-1,1)$ converge vers un ε-TC w vérifiant: $w_\lambda=\lim_{k\to\infty} u^{(k)}_\lambda=\begin{cases} x & \text{si } \lambda\in\Lambda^2_2(m-1) \\ t_1\neq x & \text{si } \lambda\in\Lambda^2_0(m-1)\cup\Gamma^0_2(m-1)\end{cases}$

En appliquant la proposition 6.2 avec l'entier $m-1\leq n-1$, on a: $w_{(2m,2)}=K(x,t_{(0,2m-2)'},t_{(0,2m-2)},y)$. Alors le fait que $t_{(0,2m-2)'}$ $t_{(0,2m-2)}$ et y sont distincts de x implique que $w_{(2m,2)}$ soit également distinct de x. Or $w_{(2m,2)}$ est la limite de la sous-suite $(u^{(k)}_{(2m,2)})_{k\in N'}$ extraite de la suite $(u^{(k)}_{(2m,2)})_{k\in\mathbb{N}}$ convergeant vers $t_{(2m,2)}=x$. D'où la contradiction.

Il s'ensuit que pour tout $m\leq n$, la suite $(u^{(k)}_{(2m,2)})$ converge vers x.

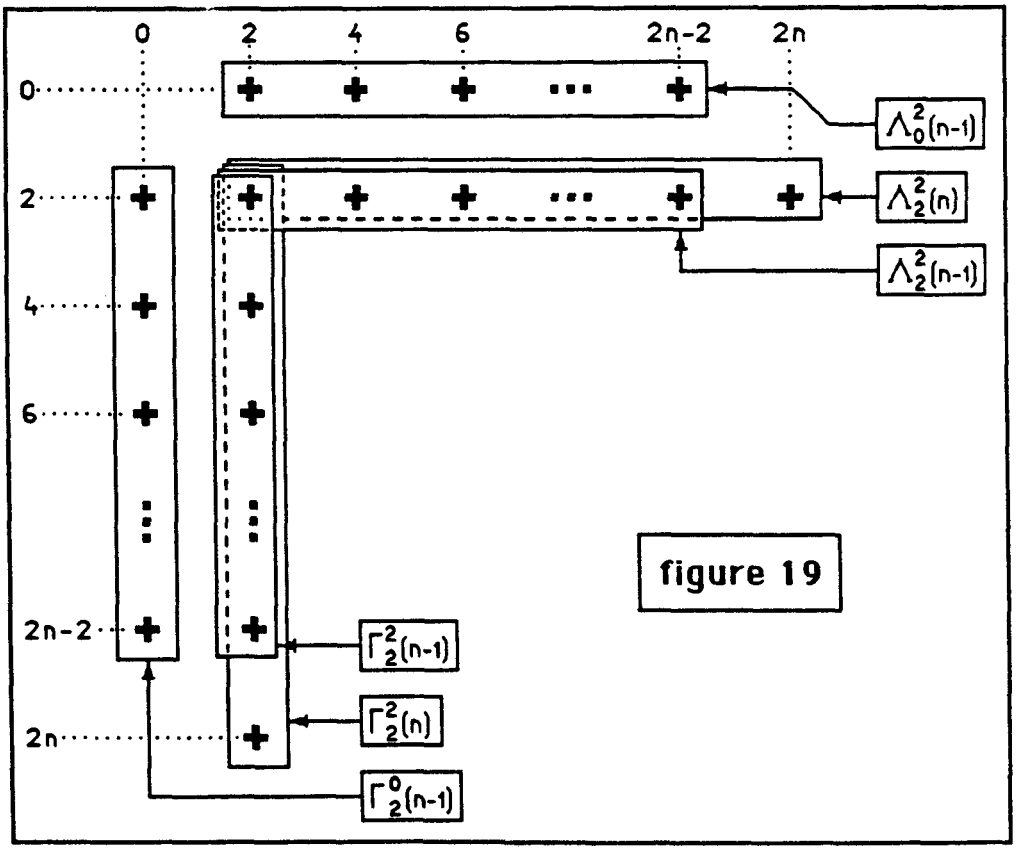


figure 19

6.4 Démonstration de la proposition 6.2

Notons que, dans le cas n=1, cette proposition ne fait qu'exprimer la continuité de l'application K. Elle est donc établie et nous supposerons désormais n≥2. Comme dans le cas précédent, nous nous contentons d'établir la proposition dans le cas où i=j=0.

D'après la proposition 4.14, $B_0^0(n,n,p)$ est une base de la grille G_n^p de bornes 2,2n+p+2,-2n,2n.

Soit une suite d'ε-TC normaux dont la restriction à $B_0^0(n,n,p)$ converge vers t. L'application de la proposition 6.1 (avec l'indice n) assure que: $\lim_{k\to\infty} u_\lambda^{(k)}=x$, $\forall\lambda\in\Lambda_2^2(n)$. De façon symétrique, on obtiendra: $\lim_{k\to\infty} u_\lambda^{(k)}=x$, $\forall\lambda\in\Lambda_{2n+2}^2(p')$, où $p'=\min(n,p+1)$.

D'après la proposition 4.15, $\bar{B}_0^0(n,n,p)=B_0^0(n,n,p)\cup\{(1,1)\}$ est une base du quinconce Q_n^p de mêmes bornes que G_n^p. Soit α_0 un élément quelconque de $\mathbb{R}^m$. A tout ε-TC $(u^{(k)})$, associons l'ε-TL $(v^{(k)})$ défini sur la base $\bar{B}_0^0(n,n,p)$ par: $v_\lambda^{(k)}=u_\lambda^{(k)}$ $\forall\lambda\in B_0^0(n,n,p)$ et $v_{(1,1)}^{(k)}=\alpha_0$.

D'après le point (iii) de la proposition 5.3, l'ε-TL $v^{(k)}$ est normal. Considérons alors la restriction de $v^{(k)}$ à la grille complémentaire $\bar{G}_n^p$ de G_n^p dans Q_n^p et appliquons à nouveau la proposition 5.3. D'après les points (i) et (ii) de cette proposition, $v^{(k)}$ est ε-TC normal.

D'après la proposition 4.14, l'ensemble $B_1^1(n-1,n-1,p)$ est une base de la grille $\tilde{G}_n^p$ incluse dans $\bar{G}_n^p$ et de bornes 4, 2n+2p+2, -2n+2, 2n-2, et il suffit de lui adjoindre les trois points (1,1), (2n+1,1) et (1,2n+1) pour obtenir une base de $\bar{G}_n^p$. Notons $w^{(k)}$ l'ε-TC restriction de $v^{(k)}$ à $\bar{G}_n^p$.

Précisons maintenant les valeurs de cet ε-TL $v^{(k)}$ sur cette base de $\bar{G}_n^p$. Après quelques calculs élémentaires utilisant la relation du losange (5.1) (toujours applicable puisque l'ε-TL $v^{(k)}$ est normal), on obtient:

$$v_{(1,2l+1)}^{(k)}=w_{(1,2l+1)}^{(k)}=\alpha_0+\sum_{\sigma=1}^{l}\left[u_{(2,2\sigma)}^{(k)}-u_{(0,2\sigma)}^{(k)}\right]^{-1} \qquad l=1,\dots,n;$$

$$v_{(2l+1,1)}^{(k)}=w_{(2l+1,1)}^{(k)}=\alpha_0+\sum_{\sigma=1}^{l}\left[u_{(2\sigma,2)}^{(k)}-u_{(2\sigma,0)}^{(k)}\right]^{-1} \qquad l=1,\dots,n;$$

$$v_{(2l+1,3)}^{(k)}=w_{(2l+1,3)}^{(k)}=w_{(2l+1,1)}^{(k)}+\left[u_{(2l+2,2)}^{(k)}-u_{(2l,2)}^{(k)}\right]^{-1} \qquad l=1,\dots,n-1;$$

$$v_{(2n+1,2l+1)}^{(k)}=w_{(2n+1,2l+1)}^{(k)}=w_{(2n+1,1)}^{(k)}+\sum_{\sigma=1}^{l}\left[u_{(2n+2,2\sigma)}^{(k)}-u_{(2n,2\sigma)}^{(k)}\right]^{-1} \qquad l=1,\dots,p;$$

Quand k tend vers l'infini, ces quantités tendent respectivement vers:

$$w_{(1,2l+1)} = \lim_{k\to\infty} w^{(k)}_{(1,2l+1)} = \alpha_0 + \sum_{\sigma=1}^{l} [x - t_{(0,2\sigma)}]^{-1} \qquad l = 1,\dots,n;$$

$$w_{(2l+1,1)} = \lim_{k\to\infty} w^{(k)}_{(2l+1,1)} = \alpha_0 + \sum_{\sigma=1}^{l} [x - t_{(2\sigma,\,0)}]^{-1} \qquad l = 1,\dots,n;$$

$$w_{(2l+1,3)} = \lim_{k\to\infty} w^{(k)}_{(2l+1,3)} = w_{(2l+1,1)} + [x - x]^{-1} = \infty \qquad l = 1,\dots,n-1;$$

$$w_{(2n+1,2l+1)} = \lim_{k\to\infty} w^{(k)}_{(2n+1,2l+1)} = w_{(2n+1,1)} + \sum_{\sigma=1}^{l} [t_{(2n+2,2\sigma)} - x]^{-1} \qquad l = 1,\dots,p.$$

Nous sommes maintenant dans les conditions d'application de la proposition 6.3 avec l'indice n-1.
En effet, la suite $w^{(k)}$ d'ε-TC normaux est telle que sa restriction à la base $B^1_1(n-1,n-1,p)$ tende vers

un ε-TC w vérifiant: $\begin{cases} w_\lambda = \infty & \forall \lambda \in \Gamma^3_3(n-1) \\ w_\lambda \neq \infty & \forall \lambda \in \Gamma^1_3(n-1) \cup \Lambda^3_1(n-1) \cup \Lambda^3_{2n+1}(n-1) \end{cases}$

D'après la proposition 6.3 (appliquée avec l'indice n-1), on a:

$$\begin{cases} \lim_{k\to\infty} w^{(k)}_\lambda = \infty \quad \forall \lambda \in \Lambda^3_3(n-1) \cup \Lambda^3_{2n-1}(p') \cup \Gamma^{2n-1}_3(p') \text{ où } p' = \min(p,n-1) \\ \lim_{k\to\infty} w^{(k)}_{(2l+1,2n+1)} = K(\infty, w_{(2n+1-2l,1)}, w_{(1,2n-2l+1)}, w_{(2n+1,2l+1)}) = w_{(1,2n-2l+1)} + w_{(2n+1,2l+1)} - w_{(2n+1-2l,1)} \\ \qquad\qquad (\text{pour } l = 1 \dots p) \end{cases}$$

Explicitons alors le second membre:

$$\lim_{k\to\infty} w^{(k)}_{(2l+1,2n+1)} = \alpha_0 + \sum_{s=1}^{n-1} [x - t_{(0,2\sigma)}]^{-1} + w_{(2n+1,1)} + \sum_{s=1}^{l} [t_{(2n+2,2\sigma)} - x]^{-1} + \alpha_0 + \sum_{s=1}^{n-1} [x - t_{(2\sigma,0)}]^{-1}$$

En particulier: $\lim_{k\to\infty} (w^{(k)}_{(2l+1,2n+1)} - w^{(k)}_{(2l-1,2n+1)}) = -[x - t_{(0,2(n-l+1))}]^{-1} + [t_{(2n+2,2l)} - x]^{-1} + [x - t_{(2(n-l+1),0)}]^{-1}$

Revenons à l'ε-TL $v^{(k)}$. Il vérifie:

$$v^{(k)}_{(2l+2,2n)} - v^{(k)}_{(2l,2n)} = (v^{(k)}_{(2l+1,2n+1)} - v^{(k)}_{(2l+1,2n-1)})^{-1} = (w^{(k)}_{(2l+1,2n+1)} - w^{(k)}_{(2l+1,2n-1)})^{-1} \quad \text{pour } l = 1,\dots,p',$$

et $(v^{(k)}_{(2l+2,2n)} - v^{(k)}_{(2l,2n)})^{-1} = w^{(k)}_{(2l+1,2n+1)} - w^{(k)}_{(2l-1,2n+1)}$ pour $l = 1,\dots,p.$

Rappelons que nous avons déjà montré que:

$$\lim_{k\to\infty} w^{(k)}_{(2l+1,2n-1)} = \infty \qquad l = 1,\dots,p'$$

$$\lim_{k\to\infty} w^{(k)}_{(2l+1,2n-1)} \neq \infty \qquad l = 1,\dots,p.$$

$$\lim_{k\to\infty} v^{(k)}_{(2,2n)} = x \quad (\text{car } (2,2n) \in \Lambda^2_2(n))$$

Nous avons alors: $v^{(k)}_{(2l+2,2n)} - v^{(k)}_{(2,2n)} = \sum_{j=1}^{l} (w^{(k)}_{(2l+1,2n+1)} - w^{(k)}_{(2l+1,2n-1)})^{-1}$

Puisque nous avons $\lim_{k\to\infty} w^{(k)}_{(2l+1,2n-1)}=\infty$ et $\lim_{k\to\infty} w^{(k)}_{(2l+1,2n-1)}=\infty$, le second membre de cette égalité tend vers 0 quand k tend vers l'infini. Il s'ensuit que: $\lim_{k\to\infty} v^{(k)}_{(2l+2,2n)}=x$ (pour $l=1,...,p'$), soit :

$$\lim_{k\to\infty} u^{(k)}_{\lambda}=x,\ \forall\lambda\in\Lambda_2^{2n}(p').$$

Enfin des relations: $(v^{(k)}_{(2l,2n+2)}-v^{(k)}_{(2l,2n)})^{-1}=w^{(k)}_{(2l+1,2n+1)}-w^{(k)}_{(2l-1,2n+1)}$ (pour $l=1,...,p'$), nous tirons:

$$\lim_{k\to\infty}(u^{(k)}_{(2l,2n+2)}-x)^{-1}=-[x-t_{(0,2(n-l+1))}]^{-1}+[t_{(2n+2,2l)}-x]^{-1}+[x-t_{(2(n-l+1),0)}]^{-1},\ l=1,...,p'$$

soit: $\lim_{k\to\infty} u^{(k)}_{(2l,\,2n+2)}=K(x,t_{(2n+2-2l,0)},t_{(0,2n+2-2l)},t_{(2n+2,2l)})$ pour $l=1,...,p'$.

6.5 Démonstration de la proposition 6.3

Soit $y\in\mathbb{R}^m$ distinct de $t_\lambda,\forall\lambda\in B_i^j(n,p,q)$. Faisons une inversion de pôle y. Nous obtenons un nouveau tableau $\bar{t}$ défini sur $B_i^j(n,p,q)$ par: $\bar{t}_\lambda=I_y(t_\lambda),\forall\lambda\in B_i^j(n,p,q)$. Ce tableau vérifie $I_y(t_\lambda)\neq\infty,\forall\lambda$ et on vérifie sans peine que l'on est dans les conditions d'application de la proposition précédente. Puisque la transformation I_y est continue (dans E_m), tout suite $u^{(k)}$ tendant vers t sur $B_i^j(n,n,p)$ est transformée en une suite $\bar{u}^{(k)}$ tendant vers $\bar{t}$ et on a donc:

$$\lim_{k\to\infty}\bar{u}^{(k)}_{\lambda}=y\quad\forall\lambda\in\Lambda_{i+2}^{j+2}(n)\cup\Lambda_{i+2n}^{j+2}(p')\cup\Gamma_{i+2}^{j+2n}(p')$$

$$\lim_{k\to\infty}\bar{u}^{(k)}_{(i+2l,j+2n+2)}=K(y,\bar{t}_{(i+2n+2-2l,j)},\bar{t}_{(i,j+2n+2-2l)},\bar{t}_{(i+2n+2,j+2l)})\ \text{pour } l=1,...,p'.$$

En utilisant à nouveau la continuité de l'inversion et le fait que l'inversion respecte l'application K, on aboutit au résultat annoncé.

6.6 Proposition 6.4

Soit t un ε-TC défini sur la base $B_i^j(n,p,q)$ avec $p+q\geq n+1, p\geq 1, q\geq 1$ et tel qu'il existe $x\in E_m$ vérifiant: $t_\lambda=x,\forall\lambda\in\Gamma_i^j(n)$ et $t_\lambda\neq x,\forall\lambda\in B_i^j(n,p,q)\setminus\Gamma_{i+2}^{j+2}(n)$.

Alors, pour toute suite $u^{(k)}$ d'ε-TC normaux dont la restriction à $B_i^j(n,p,q)$ converge vers t, on a:

$$\lim_{k\to\infty} u^{(k)}_{\lambda}=x,\quad\forall\lambda\in\Lambda_{i+2}^{j+2}(p')\cup\Lambda_{i+2n}^{j+2}(q')\cup\Gamma_{i+2+2(n-p')}^{j+2n}(p'+q'...)$$

$$\lim_{k\to\infty} u^{(k)}_{(i+2l,j+2n+2)}=K(x,t_{(i+2n+2-2l,j)},t_{(i,j+2n+2-2l)},t_{(i+2n+2,j+2l)})\ \text{pour } l=n+1-p,...,q.$$

Démonstration: Il suffit de définir un ε-TC t' sur la base $B_i^j(n,p,q)$ qui vérifie:

1° la restriction de t' à $B_i^j(n,p,q)$ est t;

2° $t'_\lambda\neq x,\ \forall\lambda\in\Lambda_i^{j+2p+2}(n-p)$.

Il est clair que tout ε-TC normal $u'^{(k)}$ sur $B'_i(n,n,q)$ a pour restriction à $B'_i(n,p,q)$ un ε-TC $u^{(k)}$, tandis que la donnée d'un ε-TC $u^{(k)}$ normal sur $B'_i(n,p,q)$ permet de construire une infinité d'ε-TC $u'^{(k)}$ normaux sur $B'_i(n,n,q)$ dont $u^{(k)}$ est la restriction et tels que $u'^{(k)}_\lambda$ soit dans un voisinage donné de t'_λ (propriété de densité). On peut alors appliquer la proposition 6.3. Le résultat est une conséquence du fait que, pour tout indice $\lambda \in \Gamma^{j+2n}_{i+2+2(n-p')}(q'+n'-n) \cup \Gamma^{j+2n+2}_{i+2+2n-p}(p+q-n)$, la limite de la suite $u'^{(k)}_\lambda$ est indépendante des valeurs auxiliaires t'_μ, pour tout μ de $\Lambda^{j+2p+2}(n)$.

7. Généralisation de l'ε-algorithme vectoriel

Dans ce chapitre nous nous proposons de montrer que la mise en œuvre de l'ε-algorithme vectoriel peut être étendue à une suite quelconque de $\mathbb{R}^m$ pourvu qu'on utilise les formules singulières introduites au chapitre précédent. Pour cela, nous montrerons que l'initialisation classique permet de construire un ε-TC unique d'un type particulier que nous qualifierons de quasi-normal. Après avoir défini les ε-TC pseudo-normaux en 7.1 et les ε-TC quasi-normaux en 7.2, nous montrons en 7.3 que la construction d'un ε-TC quasi-normal est possible dès qu'on s'est donné ses valeurs (vérifiant une certaine propriété) sur une certaine base. En 7.4 nous utiliserons les du chapitre 6 pour montrer qu'un ε-TC quasi-normal est la limite de toute suite d'ε-TC normaux dont la restriction à une base converge vers sa restriction à cette base. Il s'ensuit que les éléments définis d'un ε-TC quasi-normal sont des fonctions continues des éléments de sa restriction à la base sur laquelle il est construit. Il suffit alors de vérifier que les conditions que doivent vérifier les valeurs d'un ε-TC quasi-normal sur sa base sont réalisées lorsqu'on choisit pour valeurs celles qui correspondent à l'initialisation de l'ε-algorithme. Ceci garantit la possibilité de construire l'ε-TC quasi-normal associé à toute suite $(x_i)_{i=0\ldots n}$ finie de $\mathbb{R}^n$.

7.1 Définition des ε-TC pseudo-normaux.

Soit P une prégrille incluse dans la grille G de P_i. *Une application a définie sur P, à valeurs dans E_m, sera un ε-TC* ***pseudo-normal*** *sur G si on peut recouvrir G par un ensemble de carrés disjoints $(C_i)_{i\,(i=1,\ldots,n)}$ tels que les trois conditions suivantes soient remplies:*

1° G\P est inclus dans la réunion de l'intérieur des carrés.

2° pour i=1,...,n, a est constante sur la frontière du carré C_i; sa valeur sera notée $a_{Fr(C_i)}$

3° Pour i=1,...,n, pour chaque paire symétrique de couples de points conjugués (u,v) et (u',v') par rapport au carré C_i, nous avons:

$$a_u = K(a_{Fr(C_i)}, a_v, a_u, a_v), \text{ avec } a_u, a_v, a_u, \text{ et } a_v, \text{ distincts de } a_{Fr(C_i)}$$

La prégrille sur laquelle l'application a est définie sera appelée ***socle*** *de a.*

Remarque. L'ensemble $(C_i)_{i\,(i=1,\ldots,n)}$ des carrés associés à un ε-TC pseudo-normal est parfaitement défini par la connaissance de l'application a sur son socle. Il n'est pas nécessairement inclus dans la grille G: les carrés peuvent "déborder" de la grille. D'autre part, un ε-tableau normal est un ε-TC pseudo-normal dont tous les carrés sont de côté un.

7.2 Propriété fondamentale.

Proposition 7.1 *Soit a un ε-TC pseudo-normal sur la grille G de P_i, de socle P, et soit l'ensemble $(C_i)_{(i=1,\ldots,n)}$ des carrés de G sur la frontière desquels a est constante. Alors l'hypothèse:*

(i) il existe $(j,k) \in P_{l-i}$ et $\varepsilon \in \{-1,1\}$ tels que $u=(j+1,k+\varepsilon)$ et $v=(j-1,k-\varepsilon)$ appartiennent à P_i et $a_u=a_v$

entraîne l'exactitude de l'une des deux affirmations suivantes:

(ii) il existe $i \in \{1,\ldots,n\}$ tel que $u,v \in C_i$

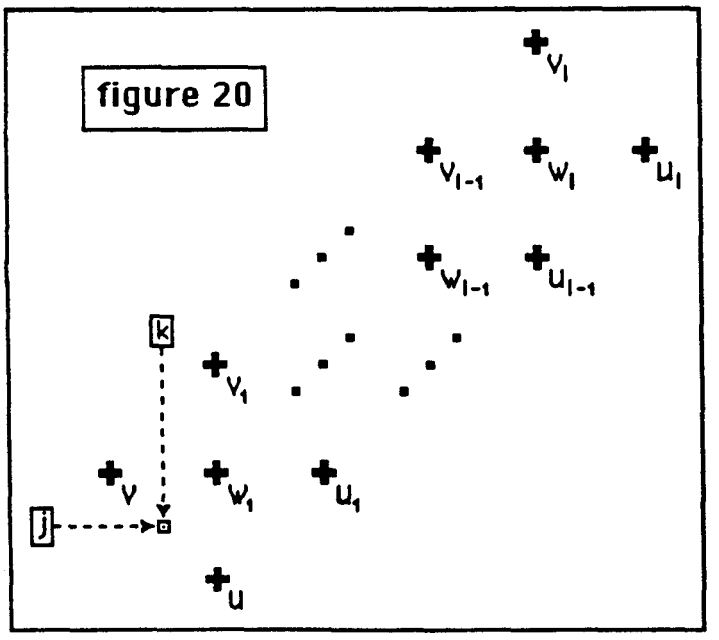
(iii) pour tout entier relatif l tel que $u_l=u+2l(1,-\varepsilon)$ appartienne à P, nous avons:

$$v_l=v+2l(1,-\varepsilon) \text{ appartient à } P \quad \text{et} \quad a_{u_l}=a_{v_l}.$$

Démonstration: Pour établir ce résultat, nous montrerons que, si (i) est réalisée tandis que (ii) ne l'est pas, alors (iii) est vérifiée.

Soit $\eta \in \{-1,1\}$. Considérons les points $u_{\eta l}=u+2l\eta(1,-\varepsilon)$ et $v_{\eta l}=v+2l\eta(1,-\varepsilon)$, pour tout l non négatif, et montrons que, pour tout l non négatif tel que $u_{\eta l}$ appartienne à P nous avons: $v_{\eta l} \in P$ et $a_{u_l}=a_{v_l}$. Pour suivre plus commodément la démonstration, le lecteur est invité à se reporter à la figure 20 dans laquelle on a supposé $\varepsilon=\eta=1$.

Soit $l \geq 1$ tel que $u_{\eta l} \in P$, ce qui implique $u_{\eta l} \in G$. Puisque G contient $u_{\eta l}$ et v, il contient $v_{\eta l}$ et $w_{\eta r}=(j,k)+(2r-1)\eta(1,-\varepsilon)$ pour $r=1,\ldots,l$. Puisque l'application a n'est pas définie à l'intérieur d'un carré, le point $w_{\eta r}$ ne peut appartenir à l'intérieur d'un carré sans que les points voisins $u_0=u$ et $v_0=v$ appartiennent à la frontière de ce carré, ce qui contredit la négation de (ii). Donc $w_{\eta r} \in P$ et a est définie en $w_{\eta r}$ et nous avons: $a_{w_\eta} \neq a_u=a_v$.



Soit $l \geq 1$ tel que $u_{\eta l} \in P$, ce qui implique $u_{\eta l} \in G$. Puisque G contient $u_{\eta l}$ et v, il contient $v_{\eta l}$ et

$w_{\eta r}=(j,k)+(2r-1)\eta(1,-\varepsilon)$ pour $r=1,\dots,l$. Puisque l'application a n'est pas définie à l'intérieur d'un carré, le point $w_{\eta r}$ ne peut appartenir à l'intérieur d'un carré sans que les points voisins $u_0=u$ et $v_0=v$ appartiennent à la frontière de ce carré, ce qui contredit la négation de (ii). Donc $w_{\eta r}\in P$ et a est définie en $w_{\eta r}$ et nous avons: $a_{w_\eta}\neq a_u=a_v$.

Considérons maintenant le carré C_p auquel appartient w_η; soit q son côté.

Si $q=1$, les points u_η et v_η sont tels que (v_η,u_0) et (u_η,v_0) sont deux couples symétriques par rapport au carré C_p (réduit à un point), et alors l'égalité $a_u=a_v$ implique $a_{u_\eta}=a_{v_\eta}$ pourvu que $u_\eta\in G$.

Si $q>1$, les points u_η, v_η, $u_{(l-1)\eta}$ et $v_{(l-1)\eta}$ appartiennent à la frontière de C_p tandis que les points $u_{r\eta}$ et $v_{r\eta}$ appartiennent à son intérieur pour $r=2,\dots,l-2$. L'application a n'est donc pas définie aux points $u_{r\eta}$ et $v_{r\eta}$ pour ces indices.

Par contre, si les points $u_{(l-1)\eta}$ et $v_{(l-1)\eta}$ sont dans G, alors: $a_{u_{(q-1)\eta}}=a_{v_{(q-1)\eta}}$. De plus, les points $u_{r\eta}$ et $v_{r\eta}$ appartiennent à la ceinture de C_p et l'égalité $a_u=a_v$ implique $a_{u_{q\eta}}=a_{v_{q\eta}}$ pourvu que $u_{q\eta}$ et $v_{q\eta}$ soient dans G.

Ainsi, nous avons mis en évidence l'existence d'un entier q strictement positif tel que, pour tout r positif inférieur ou égal à q, nous ayons:

ou bien a n'est pas défini en $u_{q\eta}$ et $v_{q\eta}$,

ou bien a est défini en $u_{q\eta}$ et $v_{q\eta}$, et alors $a_{u_{q\eta}}=a_{v_{q\eta}}$.

Si $u_{q\eta}$ n'est pas dans G, le résultat est atteint: tous les points de G de la forme $u_{q\eta}$ et $v_{q\eta}$ ont la propriété annoncée. Si $u_{q\eta}$ est dans G, on peut itérer le raisonnement à partir du couple $(u_{q\eta},v_{q\eta})$.

Corollaire. *Soit t un ε-TC pseudo-normal sur la grille G de P_l, de socle P. Alors l'hypothèse:*

(i) Il existe $(j,k)\in P_{l-i}$ et $\varepsilon\in\{-1,1\}$ tel que $u=(j+1,k+\varepsilon)$ et $v=(j-1,k-\varepsilon)$ appartiennent à P,

et $t_u\neq t_v$

entraîne que, pour tout entier relatif l tel que $u_l=u+2l(1,-\varepsilon)$ et $v_l=v+2l(1,-\varepsilon)$ appartiennent à P, nous avons l'une des deux propriétés suivantes:

(ii) ou bien: u_l et v_l appartiennent à la frontière d'un même carré.

(iii) ou bien: $t_{u_l}\neq t_{v_l}$.

Preuve: S'il existe un indice l tel que $t_{u_l}=t_{v_l}$ sans que u_l et v_l n'appartiennent à la frontière d'un même carré, la proposition précédente entraîne: ou bien $t_u=t_v$, ou bien u et v appartiennent à la frontière d'un même carré, et alors $t_u=t_v$. La négation de (i) et (ii) implique celle (i).

Définition: *Un ε-TC pseudo-normal sera dit **quasi-normal** si pour tout couple $(u=(i_u,j_u),v=(i_v,j_u))$ de points de P (socle de t) liés par: $|i_u-i_v|=|j_u-j_v|=2$, on a:*

ou bien $t_u=t_v$

ou bien u et v appartiennent à la frontière d'un même carré.

7.3 ε-TC quasi-normal associé à l'ε-algorithme.

Proposition 7.2 *Considérons la grille G_n de P_0 définie par:*

$$G_n=\{(2l,2j)\in P_0 \mid 0\le i+j\le n,\quad 0\le i-j\le n\};$$

$B_n=B_0^0(n-1,1,1)$ est donc une base simple de G_n (d'après la proposition 4.14).

Soit t une application de B_n dans $(E_m)^{card(B_n)}$ telle que:

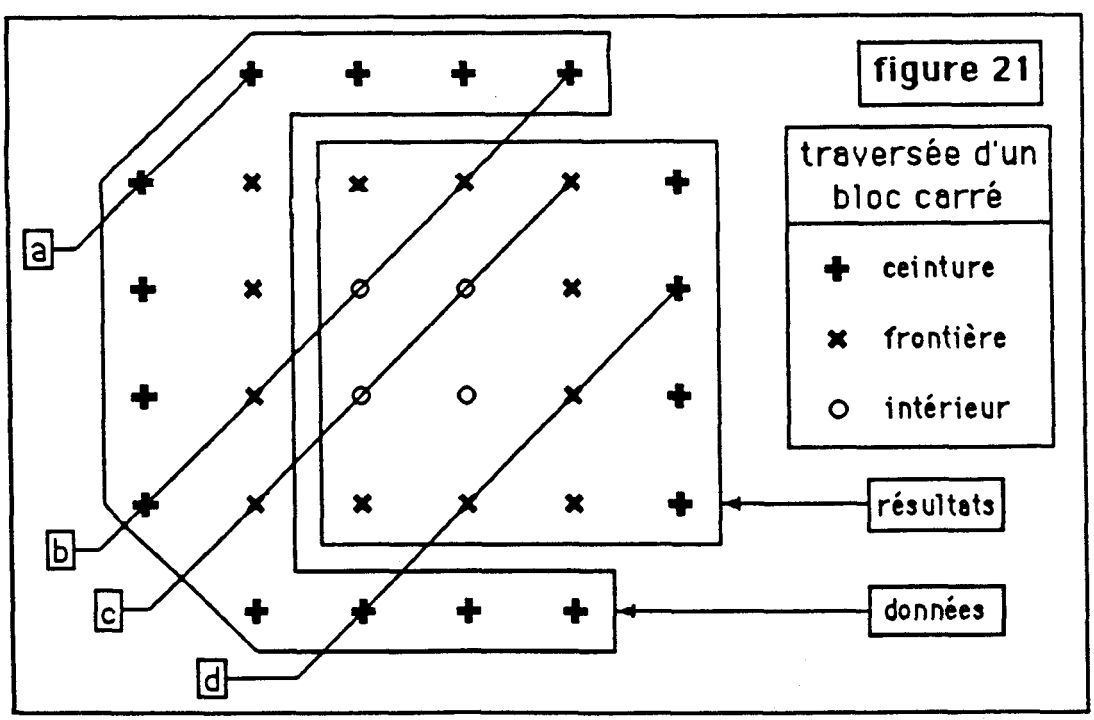
(C) pour $i=1,\dots,n-1$, $t_{(2i,-2)}$ est distinct de $t_{(2i-2,0)}$ et $t_{(2i+2,0)}$

Alors il existe un ε-TC $\tilde{t}$ quasi-normal sur G_n dont le socle P contient B_n et dont la restriction à B_n est t.

Preuve: Il suffit de construire un ε-TC répondant aux conditions. La condition (C) implique qu'un carré C_p sur la frontière duquel $\tilde{t}$ est constant soit inclus dans l'un des deux préquinconces $P_D=\{(2i,2j)\mid 0\le i\le n, 0\le j\le n\}$ ou $P_G=\{(2i,2j)\mid 1\le i\le n-1, 1\le j\le n-1\}$. Il nous suffit alors de montrer comment construire $\tilde{t}$ sur l'un de ces deux préquinconces, l'autre construction étant similaire.

Les points de $P_D\cap G_n$ sont de la forme: (2i-2j,2j) pour j=0,...,i+2 et i=0,...,n. Pour i fixé, ces points constituent une diagonale montante notée D_i. Nous déterminerons la valeur de $\tilde{t}$ sur les diagonales D_i selon l'ordre croissant de leur indice. Pour une diagonale fixée, nous respecterons l'ordre défini par les j croissant (correspondant à la mise en œuvre classique de l'ε-algorithme).

Une diagonale coupe un carré fixé en zéro, un ou plusieurs points. Par exemple, sur la figure 21, la diagonale (a) coupera le carré suivant deux points de sa ceinture, (b) le coupera suivant deux points de sa ceinture, deux points de sa frontiére et un point intérieur, (c) suivant deux points de sa frontière et deux points intérieurs et (d) suivant deux points de sa frontière et deux points de sa ceinture.



Le principe de la construction consistera à définir simultanément les valeurs de $\tilde{t}$ sur

l'intersection d'une diagonale avec un carré. En effet, la taille d'un carré est connue dès que le nombre de valeurs qui figurent sur la première colonne de sa frontière est connue: on peut alors construire toute la frontière du carré. Les valeurs de la dernière colonne de la ceinture dépendent des autres valeurs de la ceinture et de la valeur sur la frontière par l'intermédiaire de la fonction K. Il est clair que ces valeurs dont dépendent la dernière colonne ont été antérieurement calculées si nous respectons l'ordre défini plus haut. Notons enfin que, grace au corollaire de la proposition 7.1, la condition (C) nous assure de toujours être dans les conditions de construction des points de la dernière colonne. Nous pouvons alors définir la construction par:

pour i=2,...,n faire --construction de la diagonale D_i

  pour j=1,...,n+2 faire --construction du j^e élément de cette diagonale

    si $\tilde{t}_{(2(i-j),2(j-1))} = \tilde{t}_{(2(i-j-1),2(j-2))}$ alors

    -- nous sommes en dessous d'un carré singulier dont la taille a déjà été déterminée:
      détermination de la position de l'intersection de la diagonale avec la dernière colonne de la frontière et celle de la ceinture du carré, et actualisation de l'indice j.

    sinon si $\tilde{t}_{(2(i-j),2(j-1))} = \tilde{t}_{(2(i-j-1),2(j-1))}$ alors

    --nous traversons un carré dont la taille n'a pas encore été déterminée:
      détermination du terme de la frontière (première ligne du carré) et actualisation de l'indice j;

    sinon --cas normal:
      détermination de la valeur du quatrième élément de la ceinture du carré courant et j progresse d'une unité.

    fsi;

  fpour j;

fpour i;

7.4 Limite d'ε-TC normaux.

Proposition 7.3 *Soit t un ε-TC pseudo-normal sur $G_{n'}$ de socle P contenant B_n.*

Pour toute suite $(t)^{(i)}_{(i\in IN)}$ d'ε-TC normaux vérifiant: $\lim_{i\to\infty} t^{(i)}_\lambda = t_\lambda$, $\forall \lambda \in B_{n'}$

nous avons: $\lim_{i\to\infty} t^{(i)}_\mu = t_\mu$, $\forall \mu \in P$.

Preuve: Considérons un carré (C_l) de G_n (non nécessairement contenu tout entier dans G_n) sur la frontière duquel t est constant. Par définition d'un ε-TC pseudo-normal, nous avons: $t_\lambda \neq t_{fr(C_l)}$, pour tout λ de la ceinture de C_l. Soient (u,v) et (r,s) deux couples symétriques de points contenus dans P conjugués par rapport au carré C_l et supposons que, selon l'ordre défini au paragraphe précédent, t_s se calcule partir de t_r, t_u, t_v et $t_{fr(C_l)}$. D'après la proposition 6.4, nous avons:

$$\lim_{i\to\infty} t^{(i)}_u = t_u,\ \lim_{i\to\infty} t^{(i)}_v = t_v \text{ et } \lim_{i\to\infty} t^{(i)}_r = t_r \text{ implique: } \lim_{i\to\infty} t^{(i)}_s = t_s.$$

En respectant l'ordre défini antérieurement pour le calcul des éléments d'un ε-TC quasi-normal à partir de sa valeur sur la base $B_{n'}$ on montre ainsi de proche en proche que: $\lim_{i\to\infty} t^{(i)}_\mu = t_\mu$, $\forall \mu \in P$.

7.5 Continuité

Proposition 7.4 *Soit μ appartenant à la grille G_n de base B_n. Pour tout ε-TC t défini sur B_n vérifiant (C), l'ε-TC $\tilde{t}$ pseudo-normal sur G_n et dont la restriction à B_n est t vérifie:*

ou bien t_μ n'est pas défini (car $\mu \notin P$, socle de $\tilde{t}$);

ou bien t_μ est une fonction continue de $t_\lambda, \forall \lambda \in B_n$.

Preuve: Il suffit de noter que la densité des ε-TC normaux dans l'ensemble des ε-tableaux et la compacité de E_m nous permettent d'appliquer le théorème de prolongement par continuité à l'application qui, à tout t défini sur B_n et vérifiant (C) associe la valeur t_μ. La condition $\lim_{i\to\infty} t_\mu^{(i)} = t_\mu$, pour toute suite $t^{(i)}$ d'ε-TC normaux dont la restriction à B_n converge t, est assurée par la proposition précédente, pourvu que $\mu \in P$, socle de t.

7.6 Extension de l'ε-algorithme vectoriel.

Théorème: *Soient n+1 éléments $(x_i)_{(i=0,\dots,n)}$ de R^m;*
Si t est un ε-TC défini sur B_n par:

$$t_{(2i,-2)} = \infty, \forall i = 1,\dots,n-1 \quad \text{et} \quad t_{(2i,0)} = x_i, \quad \forall i = 0,\dots,n \ .$$

Alors

1° il existe un ε-TC quasi-normal $\tilde{t}$ dont la restriction à B_n est t.

2° tout élément défini de cet ε-TC est une fonction continue des valeurs x_i.

Démonstration: Il suffit de vérifier que t est vérifie les conditions (C), ce qui est immédiat.

7.7 Interprétation de l'extension de l'ε-algorithme vectoriel.

Revenons maintenant à l'indiçage classique de l'ε-algorithme pour interpréter la signification de cette extension de l'ε-algorithme vectoriel. Nous venons de voir que, si nous initialisons $\varepsilon_{-2}^{(n)} = \infty$, $n \geq 2$ et $\varepsilon_0^{(n)} = x_n$, $n \geq 2$, il est possible de construire un certain ε-tableau (même si plusieurs valeurs consécutives d'une même colonne sont égales), mais ce tableau peut comporter des "trous" de forme carrée à l'intérieur desquels la valeur du tableau n'est pas définie.

Comme dans le chapitre 3, nous notons $\varepsilon_{2k}^{(n)}(x_i)$ l'élément d'indice (2k,n) construit à partir de la suite $(x_i)_{i\geq 0}$ et dont le calcul fait appel aux 2k+1 valeurs consécutives de la suite: $x_n, x_{n+1},\dots, x_{n+2k}$. Nous distinguerons au moyen d'un ~ les éléments construits au moyen des règles étendues. Il est clair que, si $\varepsilon_{2k}^{(n)}(x_i)$ existe, nous avons $\tilde{\varepsilon}_{2k}^{(n)}(x_i) = \varepsilon_{2k}^{(n)}(x_i)$. La fonction $\tilde{\varepsilon}_{2k}^{(n)}(x_i)$ est définie et continue au voisinage de tout (2k+1)-uplet $(x_n, x_{n+1},\dots, x_{n+2k})$ pour lequel l'indice (2k,n) n'est pas dans un "trou". Soit (x_i) une suite contenant un tel (2k+1)-uplet. Grace à la continuité de la fonction $\tilde{\varepsilon}_{2k}^{(n)}(x_i)$, pour toute suite de suites $(x_i^l)_{l\geq 0}$ tendant vers la suite (x_i), nous avons:

$$\lim_{l\to\infty} \tilde{\varepsilon}_{2k}^{(n)}(x_i^l) = \varepsilon_{2k}^{(n)}(x_i)$$

Par contre, dans les "trous", là où $\varepsilon_{2k}^{(n)}$ n'est pas défini,il n'y a rien de tel. On peut même montrer que, si la suite (x_i) est telle que $\tilde{\varepsilon}_{2k}^{(n)}(x_i)$ ne soit pas défini pour un indice (k,n) fixé, pour tout $\alpha \in E_m$, on peut choisir une suite de suites $(x_i^l)_{l\geq 0}$ tendant vers la suite (x_i) et telle que:

$$\lim_{l\to\infty} \tilde{\varepsilon}_{2k}^{(n)}(x_i^l) = \alpha.$$

Cela signifie que l'application $(x_i)\to\tilde{\varepsilon}_{2k}^{(n)}(x_i)$ ne peut pas être définie par continuité en un tel point: l'extention que nous avons obtenue est donc maximale. Pour k et n fixés, l'application $\tilde{\varepsilon}_{2k}^{(n)}(x_i)$ est l'extension continue maximale de l'application $\varepsilon_{2k}^{(n)}(x_i)$ sur les (2k+1)-uplets $(x_n, x_{n+1},..., x_{n+2k})$.

Ce résultat reste pourtant imprécis parce que, dans le cas général de l'ε-algorithme vectoriel, nous ne savons pas caractériser précisément la transformation $\varepsilon_{2k}^{(n)}(x_i)$. Par contre, dans le cas de l'ε-algorithme scalaire, nous avons un résultat plus précis puisque nous savons que, lorsqu'on peut le calculer, le terme $\varepsilon_{2k}^{(n)}(x_i)$ s'identifie avec le transformé de Shanks [10]:

$$e_k(x_n)=H_{k+1}(x_n)/H_k(\Delta^2 x_n),$$

où $H_l(t_n)$ est le déterminant de Hankel défini par:

$$H_l(t_n)=\text{dét}\begin{pmatrix} t_n & \cdots & t_{n+l-1} \\ \vdots & & \vdots \\ t_{n+l-1} & \cdots & t_{n+2(l-1)} \end{pmatrix}$$

En effet, l'ε-algorithme scalaire est une méthode récursive permettant de calculer $e_k(x_n)$ sans nécessiter le calcul des déterminants dont il est le quotient. Naturellement, la transformation n'est pas définie quand ces deux déterminants sont simultanément nuls. Toutefois, la valeur $e_k(x_n)$ peut fort bien être définie alors que la valeur $\varepsilon_{2k}^{(n)}(x_i)$ ne l'est pas: cela arrive si les relations de récurrence utilisées dans l'ε-algorithme font appel à des valeurs intermédiaires indéterminées, ce qui correspond au cas où plusieurs valeurs consécutives d'une même colonne sont égales. Pour k et n fixés, la fonction $e_k(x_n)$ est donc un prolongement par continuité dans E_m de la fonction $\varepsilon_{2k}^{(n)}(x_i)$. L'unicité du prolongement par continuité maximal dans un espace topologique régulier implique donc que $\tilde{\varepsilon}_{2k}^{(n)}(x_i)$ soit encore un prolongement par continuité de $e_k(x_n)$. En fait, une analyse plus approfondie que nous développerons pas ici montre que, dans ce cas scalaire, il y a identité des deux notions: l'algorithme récursif que nous avons présenté permet donc de calculer tous les transformés de Shanks $e_k(x_n)$ qui sont définis. C'est donc une extension totalement fiable (reliable en anglais) de l'ε-algorithme.

Références

1. **Baker G.A. Jr.**- Recursive calculation of Padé approximants. Padé approximants and their applications, P.R. Graves-Morris ed., Academic press, London (1973) 83-91.

2. **Brezinski C.**- Méthodes d'accélération de la convergence en analyse numérique. Thèse d'Etat, Grenoble (1971).

3. **Brezinski C.**- Sur un algorithme de résolution des systèmes non liné-aires. C.R.A.S. Paris 272A (1971) 145-148.

4. **Brezinski C.**- Some results in the theory of the vector ε-algorithm. Linear Algebra 8 (1974) 77-86.

5. **Brezinski C.**- Computation of the eigenelements of a matrix by the ε-algorithm. Linear Algebra 11 (1975) 7-20.

6. **Brezinski C.**- Accélération de la convergence en analyse numérique. Lecture Notes in Math. 584, Springer Verlag, Berlin Heidelberg New-York (1977).

7. **Cordellier F.**- Interprétation géométrique d'une étape de l'ε-algorithme. Publication40, Laboratoire de calcul,Université de Lille I (1973).

8. **Cordellier F.**- Particular rules for the vector ε-algorithm.Numer. Math. 27 (1977) pp 203-207.

9. **Cordellier F.**- Démonstration algébrique de l' extension de l' identité de Wynn aux tables de Padé non normales. L.N.M. 765 (1979) 36-60.

10. **Cuyt A.**- Singular rules for the calculation of non-normal multivariate Padé approximants. J. Comp.Appl.Math. 14 (1986) 289-301.

11. **Gragg W.B.**- The Padé table and its relation to certain algorithms of numerical analysis. S.I.A.M. Review 14 (1962) 1-62.

12. **Graves-Morris P.R. and Jenkins C.D.**- Vector-valued rational interpolants III. Constr. Approx. 2 (1986) 263-289.

13. **Graves-Morris P.R. and Jenkins C.D.**- Degeneracies of generalised inverse, vector-valued approximants. A paraître dans Constr. Approx.

14. **Graves-Morris P.R.**- The numerical calculation of Padé approximants. Padé approximants and its applications, L. Wuytack ed., Lecture Notes in Math. 765, Springer Verlag, Berlin Heidelberg New-York (1979) 231-245.

15. **Hadamard J.**- Récents progrès de la géométrie anallagmatique. Œuvres Tome 2, Editions du CNRS (1968) 937-986.

16. **Mc Leod J.B.**- A note on the ε-algorithm. Computing 7 (1971) 17-24.

17. **Shanks D.**- Nonlinear transformations of divergent and slowly convergent sequences. J. Math.Phys. 34 (1955) 1-42.

18. **Wynn P.**- On a device for computing the $e_m(S_n)$ transformation. M.T.A.C. 10 (1956) 91-96.

19. **Wynn P.**- L'ε-algorithma e la tavola di Padé. Rend. di Mat. Roma 20 (1961) 403-408.

20. **Wynn P.**- Acceleration techniques for iterated vector and matrix problems. Math. Comp. 16 (1962) 301-322.

21. Wynn P.- Singular rules for certain nonlinear algorithms. B.I.T. 3 (1963) 175-195.

22. Wynn P.- Upon systems of recursions which obtain among the quotients of the Padé table. Numer. Math. 8 (1966) 264-269.

23. Wynn P.- A numerical method for estimating parameters in mathematical models. Centre de Recherches Mathématiques, Université de Montréal, CRM-443 (1974).

24. Wynn P.-Acceleration techniques in numerical analysis, with particular reference to problems in oner independent variable. Proceedings IFIP Congress 1962, North Holland Pub. Co. (1963) 149-156.

Annexe A: programme de mise en œuvre

A.1 présentation sommaire.

On propose ici une version en TURBO PASCAL pour MacIntosh de l'ε-algorithme intégrant les règles singulières généralisées décrites dans le papier. Cet algorithme utilise la technique du losange mobile introduite par Wynn [24] qui permet de décrire un algorithme de losange en n'utilisant qu'un
tableau unidimensionnel dont la taille est le nombre de termes de la suite qu'on veut transformer. Pour décrire la traversée des carrés singuliers, nous avons introduit un tableau unidimensionnel d'indices repérant la taille des singularités rencontrées. Ce tableau l(0:n) est tel que, la valeur effective de l'élément correspondant à l'indice j dans la diagonale montante numéro i est donnée par a(j-l(j)). En effet, le tableau a(0:n) ne contient la valeur des éléments de la diagonale traitée que dans le cas où on celle-ci ne traverse aucun bloc carré singulier. Dans les autres cas, on profite de ce qu'il suffit de mémoriser une seule fois la valeur commune à tous les éléments de la frontière et de l'intérieur de chaque carré pour utiliser les places ainsi rendues disponibles à la mémorisation des valeurs des colonnes antérieures utilisées dans la mise en œuvre des règles singulières. Nous ne décrirons pas ici plus précisément cette technique qui fera l'objet d'une publication ultérieure, mais le lecteur intéressé peut déjà suivre le fonctionnement de cette technique au moyen de l'exemple qui illustre la mise en œuvre de ce programme.

A.2 programme Pascal.

```
program epsalgosing;
{**************************************************************************
*   Mise en œuvre de l'epsilon-algorithme avec règles singulières généralisées et mémorisation
*   d'une seule diagonale montante.
*
**************************************************************************
*  DECLARATIONS(indications sommaires):
*  x: contient les termes de la suite à laquelle on applique l'algorithme;
*  a: contient les valeurs qu'il est essentiel de mémoriser lorsqu'on passe d'une diagonale à la sui-
*     vante; ces valeurs peuvent concerner des éléments situés bien en deçà de la diagonale actuel-
*     lement traitée dans le cas où l'emploi des règles singulières est indispensable.
*  l: est un tableau indiquant la taille de la singularité qu'on a rencontrée; ce tableau est tel que la
*   valeur de l'élément qui se trouve à l'indice j de la diagonale courante est donnée par: a(j-l(j)).
**************************************************************************}
type indice=0..30;
   triplet =record index:integer;
                   val1,val2:real
            end;
var  i,j,k,n,m: integer;
   r,s,t,p,y:real;
   b: boolean;
   l: array [indice] of integer;
   a,x: array [indice] of real;
    res:triplet;
```

```
   label 1;
begin
  writeln ('nombre de données');
  readln ( n);
  for i:= 0 to n do readln (x[i]);
  for i:= 0 to n do writeln (x[i]);
  a[0]:=x[0]; l[0]:=0;
  for i:=1 to n do
  begin  {TRAITEMENT D'UNE DIAGONALE MONTANTE}
    r:=0.0; t:=x[i]; k:=i; m:=0; b:=FALSE; j:=k-1;
    while (j>=0) do
    begin {TRAVERSEE D'UN BLOC CARRE}
      s:=a[j-l[j]];
      if (s<>t) then
      begin { CAS REGULIER}
        l[k]:=m;
        if (m=0) then a[k]:=t else a[k]:=r;
        p:=r+1/(t-s); b:=FALSE; k:=j; m:=0; r:=a[k]; t:=p;
      end { FIN CAS REGULIER}
      else
      begin {OUVERTURE DE SINGULARITE}
        l[k]:=l[j]+1;
        if (b) then a[k]:=y else a[k]:=r;
        m:=-l[k]; k:=k+2*m;
        if (k>=0) then begin r:=a[k];
                             b:=TRUE
                       end
                  else goto 1;
      end { OUVERTURE DE SINGULARITE };
      if (l[k]<>0) then
      begin { FERMETURE DE SINGULARITE}
        j:=k-1;
        if (j>=0) then
        begin l[k]:=m; y:=a[j-l[j]];
          if (b) then a[k]:=y else a[k]:=p;
           k:=j-2*l[j]-1;
          if (k>=0) then
          begin s:=a[k]; p:=s+t-r; m:=0; r:=y; t:=p; b:=TRUE end
          else goto 1
        end
        else goto 1;
      end { FIN FERMETURE DE SINGULARITE};
      j:=k-1
    end {   FIN DE LA TRAVERSEE D'UN BLOC CARRE};
    l[k]:=m;
    if (m=0) then a[k]:=t;
    {IMPRESSION DE LA DIAGONALE MONTANTE}
1:    writeln ("i= ",i);
      for j:=i downto 0 do write (l[j]); writeln;
      for j:=i downto 0 do write (a[j]); writeln;
      for j:=i downto 0 do write (a[j-l[j]]); writeln;
  end;
  readln
end.
```

A.3 Exemple de mise en œuvre.

On a choisi une suite x satisfaisant une relation de récurrence d'ordre 4: $x_i = 3 * x_{i-4}$, pour tout $i \geq 4$., avec l'initialisation: $x_0 = x_1 = x_2 = 1$ et $x_3 = 2$. On sait alors [4] que la huitième colonne de l'ε-tableau est identique-ment nulle. Le tableau qui suit permet de suivre le déroulement de l'algorithme: pour chaque diagonale mon-tante, on affiche trois vecteurs: le vecteur l qui indique la nature et la localisation des singularités rencontrées, le vecteur a qui contient toutes les valeurs utiles à la mise en œuvre ultérieure des règles singulières, et le vecteur b dont la composante d'indice j est donnée par $b_j = a_{j-l_j}$, et qui contient la valeur effective de l'élément d'indice j de la diagonale avec toutefois la restriction suivante: toute les valeurs intérieures à un carré singulier sont forcées à la même valeur, celle de la frontière de ce carré.

A.4 Affichage des résultats.

```
  0    1    2    3    4    5    6    7    8    9   10   11   12   13   14
i=1
  1    0
 0.00 1.00
 1.00 1.00
  2    1    0
 0.00 0.00 1.00
 1.00 1.00 1.00
  0    0    1    0
 2.00 1.00 0.00 1.00
 2.00 1.00 1.00 1.00
  0    1    0   -1    0
 3.00 2.00 1.00 1.00 1.00
 3.00 1.00 1.00 1.00 1.00
i=5
  1    0   -1    0    0    0
 0.00 3.00 1.00 1.00 2.00 2.00
 3.00 3.00 3.00 1.00 2.00 2.00
  2    1    0   -1   -2    1    0
 0.00 9.00 3.00 1.00 1.00 2.00 2.00
 3.00 3.00 3.00 3.00 3.00 2.00 2.00
  0    0    1    0   -1    0    0    0
 6.00 0.33 0.00 3.00 1.00 1.33 1.50 2.00
 6.00 0.33 3.00 3.00 3.00 1.33 1.50 2.00
  0    1    0   -1    0    1    0   -1    0
 9.00 6.00 0.33 3.00 3.00 3.00 1.33 1.50 0.00
 9.00 0.33 0.33 0.33 3.00 1.33 1.33 1.33 0.00
i=9
  1    0   -1    0    0    0    0    0    0    0
 0.00 9.00 0.33 0.33 6.00 0.67 1.50 1.33 0.00 ***
 9.00 9.00 9.00 0.33 6.00 0.67 1.50 1.33 0.00 ***
  2    1    0   -1   -2    1    0   -1    1    0   -1
 0.00 0.00 9.00 0.33 0.33 6.00 0.67 1.50 1.33 0.00 ***
 9.00 9.00 9.00 9.00 9.00 0.67 0.67 0.67 0.00 0.00 0.00
  0    0    1    0   -1    0    0    0    2    1    0   -1
18.00 0.11 0.00 9.00 0.33 0.44 4.50 0.67 0.67 1.33 0.00 ***
18.00 0.11 9.00 9.00 9.00 0.44 4.50 0.67 0.00 0.00 0.00 0.00
  0    1    0   -1    0    1    0   -1    3    2    1    0   -1
27.00 18.00 0.11 9.00 9.00 9.00 0.44 4.50 0.67 0.67 1.33 0.00 ***
27.00 0.11 0.11 0.11 9.00 0.44 0.44 0.44 0.00 0.00 0.00 0.00 0.00
i=13
  1    0   -1    0    0    0    0    0    4    3    2    1    0   -1
 0.00 27.00 0.11 0.11 18.00 0.22 4.50 0.44 0.44 0.67 0.67 1.33 0.00 ***
27.00 27.00 27.00 0.11 18.00 0.22 4.50 0.44 0.00 0.00 0.00 0.00 0.00 0.00
  2    1    0   -1   -2    1    0   -1    5    4    3    2    1    0   -1
 0.00 0.00 27.00 0.11 0.11 18.00 0.22 4.50 0.44 0.44 0.67 0.67 1.33 0.00 ***
27.00 27.00 27.00 27.00 27.00 0.22 0.22 0.22 0.00 0.00 0.00 0.00 0.00 0.00 0.00
```

A.5 Visualisation des résultats affichés.

La figure 22 présente le tableau construit à partir des 15 premières valeurs de la suite x. On a choi-si 2 diagonales montantes pour illustrer le fonctionnement de l'algorithme. Le lecteur vérifiera que leurs valeurs sont celles du tableau b, (construit à partir des tableaux l et a), à l'exception des va-leurs intérieures aux carrés singuliers. D'autre part, nous avons repéré en les encerclant les va-leurs qui, pour chacune de ces diagonales, seront ultérieurement utilisées dans la mise en œuvre des règles singulières. Ces valeurs sont celles du tableau a lors du calcul des éléments de cette diagonale.

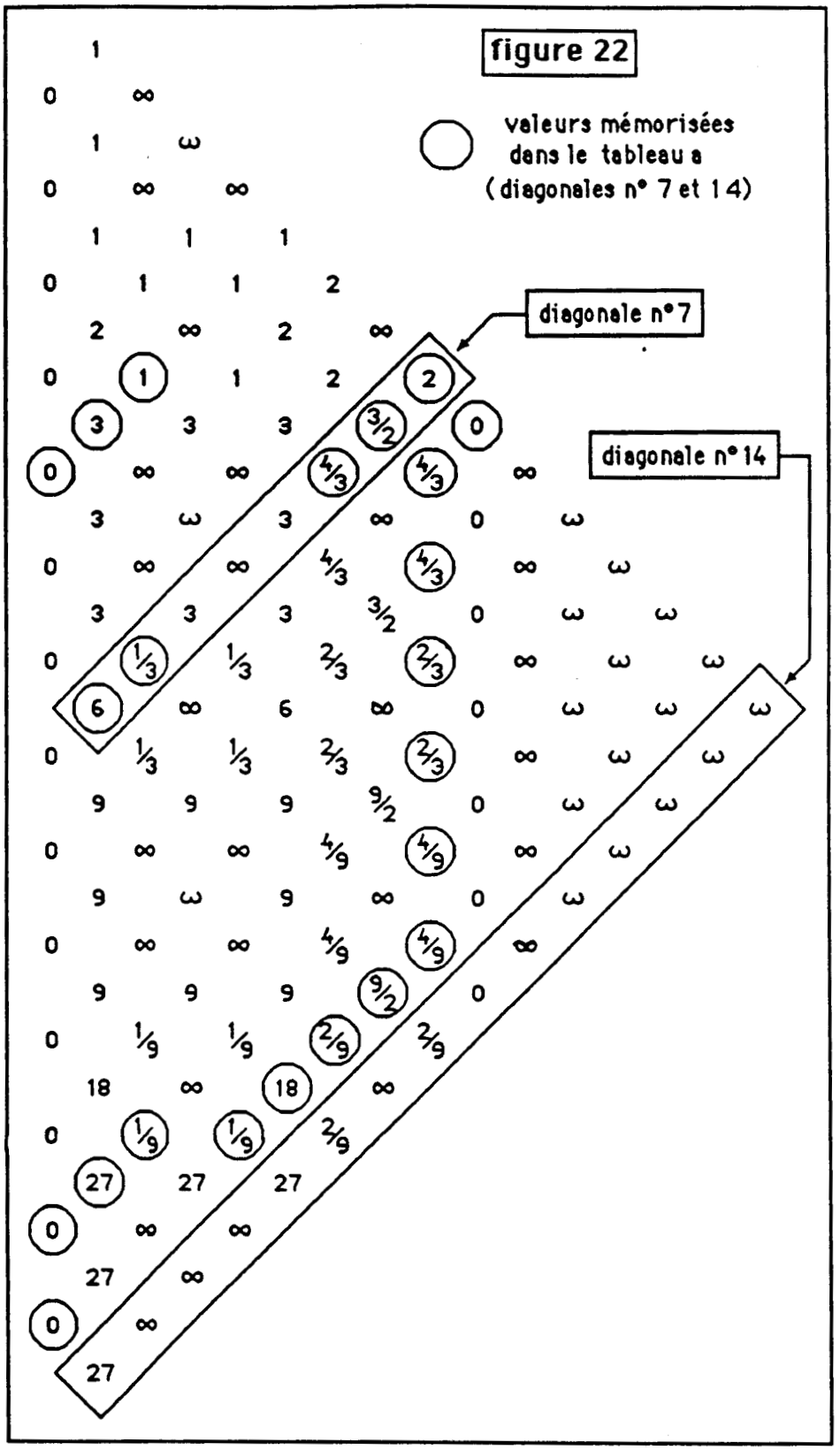


figure 22
valeurs mémorisées dans le tableau a (diagonales n° 7 et 14)
diagonale n°7
diagonale n°14

Annexe B: extension de l'identité de Wynn

(schéma de démonstration)

On se contente ici de donner de brèves indications sur un schéma de démons-tration de l'extension de l'identité de Wynn aux tables de Padé non normales.

B.1- On sait [11] que, pour des tables de Padé normales, les approximants sont égaux à l'intérieur de blocs carrés.

B.2- Les règles singulières définies sur l'ε-algorithme vectoriel sont encore valables pour l'ε-algorithme scalaire. Elles permettent construire la table de Padé pour une valeur fixée de la variable, sauf à l'intérieur des carrés singuliers. En raison de la continuité de la valeur des éléments non intérieurs aux carrés singuliers par rapport à celle des éléments de la première colonne, on a encore $e_k(x_n) = H_{k+1}(x_n)/H_k(\Delta^2 x_n)$, ce qui montre que les valeurs calculées au moyen des règles singulières correspondent aux valeurs de la table de Padé, du moins dans la partie inférieure de cette table située en dessous de la diagonale; mais une variante de l'ε-algorithme scalaire obtenue en complétant l'initialisation permet de calculer la valeur ponctuelle des autres approximants.

B.3- Compte tenu de 1, pour l'ensemble des approximants de Padé d'une fonction f dont la somme des degrés est inférieure à un entier n donné, il n'existe qu'un nombre fini de structures différentes comportant des carrés. Par exemple, pour n=2, les seules structures possibles sont les suivantes:

figure 23 — structure de la sous-table des approximants de Padé dont la somme des degrés est bornée par 2

	0	0	0		0	0	0		0	0	0		0	0	0		0	0	0
∞	P	Q	R	∞	P	P	R	∞	P	Q	Q	∞	P	Q	R	∞	P	P	P
∞	S	T		∞	P	P		∞	S	Q		∞	S	S		∞	P	P	
∞	U			∞	U			∞	U			∞	S			∞	P		
table normale				tables non normales															

B.4- Une table de Padé limitée aux approximants dont la somme des degrés est bornée par n ayant une structure donnée, la table associée à une valeur de la variable aura soit la même structure, soit une structure avec des blocs carrés plus grands (constitués par la réunion de blocs carrés de la table), mais ceci ne peut arriver que lorsque deux fractions rationnelles prennent la même valeur, c'est à dire pour un nombre fini de valeurs de la variable. Notons X cet ensemble fini de valeurs de la variable tel que deux approximants distincts de notre table de Padé limitée prennent la même valeur.

B.5- Pour toute valeur $x \notin X$, la table de Padé ponctuelle aura la même structure que la table de Padé. D'après 2, nous savons construire cette table de Padé ponctuelle et la relation de la croix est valide autour de tout carré singulier: à chacune de ces relations, on peut associer le résidu:

$$R(t) = (N_i(t)-P(t))^{-1}+(S_{p+1-i}-P(t))^{-1}-(W_i(t)-P(t))^{-1}+(E_{p+1-i}-P(t))^{-1}$$

C'est une fraction rationnelle nulle sur un ensemble infini de points, donc identiquement nulle: l'identité de Wynn s'étend donc aux table de Padé non normales.

UNE MISE EN OEUVRE NUMÉRIQUEMENT STABLE DE L'ε-ALGORITHME VECTORIEL

Florent CORDELLIER

Université de Lille I
I.E.E.A. Informatique
B.P. 36
59650 VILLENEUVE D'ASCQ

La mise en oeuvre de l'ε-algorithme (scalaire ou vectoriel) conduit souvent à des erreurs de calcul notables qui nuisent beaucoup à son efficacité. Pour l'ε-algorithme scalaire, Wynn [6] a proposé une règle particulière permettant de diminuer de façon sensible les erreurs de calcul dans le cas où deux valeurs consécutives d'une même colonne deviennent voisines. J'ai proposé une règle analogue [2] pour l'ε-algorithme vectoriel, et l'introduction d'une technique de contrôle des erreurs de calcul dans les calculs vectoriels m'a permis de justifier des critères d'application de cette règle [3]. Wynn [6] a également montré que, si deux termes consécutifs d'une même colonne sont égaux, la mise en oeuvre de l'ε-algorithme scalaire peut être poursuivie moyennant l'introduction d'une règle singulière. Ce résultat a récemment été étendu [1] à l'ε-algorithme vectoriel dans le cas où n vecteurs consécutifs d'une même colonne sont égaux.

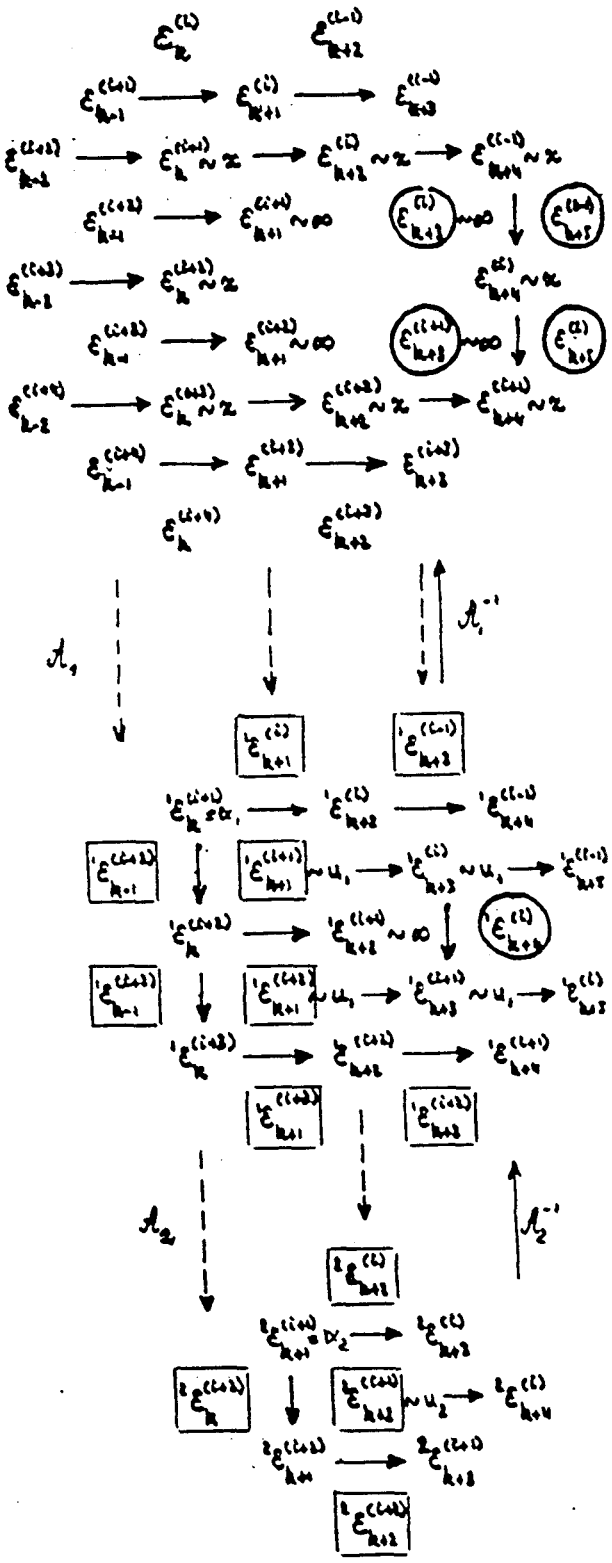
L'algorithme qui est présenté ici jouit des deux propriétés suivantes :

- D'une part, il propose un traitement numérique stable dans le cas où n vecteurs consécutifs d'une même colonne sont "voisins" à condition qu'ils soient "encadrés" par des vecteurs "non voisins".

- D'autre part, il assure la compatibilité de ces règles avec les règles singulières évoquées plus haut : ces règles singulières deviennent alors les cas limites des nouvelles règles.

La mise en oeuvre de cette variante de l'ε-algorithme vectoriel repose sur le fait que les relations qui lient les éléments des colonnes de même parité dans le tableau de l'ε-algorithme sont conservées si on prend les transformés anallagmatiques [4] des éléments en question.

Bien que ces règles soient utilisables pour n quelconque , nous nous contentons de les présenter dans ce résumé dans le cas où n=3 sans rappeler les notations classiques introduites par Wynn [5]. Précisons toutefois la règle du losange :
$\varepsilon_{k+1}^{(i)} - \varepsilon_{k-1}^{(i+1)} = (\varepsilon_k^{(i+1)} - \varepsilon_k^{(i)})^{-1}$ où $y^{-1} = y/||y||^2$ est l'inverse géométrique de y.
Cette règle permet de calculer l'élément situé à l'un des sommets d'un losange dès qu'on connaît les 3 autres. Dans les schémas qui suivent, une flèche indiquera l'élément calculé.



Les entiers k et i étant fixés, on suppose que les 3 vecteurs $\epsilon_k^{(i+j)}$ soient "voisins" de $x \in R^m$ pour j=1,2,3 alors que les points $\epsilon_{k-2}^{(i+j+1)}$ (j=1,2,3), $\epsilon_k^{(i)}$, $\epsilon_k^{(i+4)}$, $\epsilon_{k+2}^{(i-1)}$ et $\epsilon_{k+2}^{(i+3)}$ en sont "éloignés". Alors $\epsilon_{k+1}^{(i+1)}$ et $\epsilon_{k+1}^{(i+2)}$ sont "voisins" de ∞ tandis que $\epsilon_{k-1}^{(i+2)}$, $\epsilon_{k-1}^{(i+3)}$, $\epsilon_{k+1}^{(i)}$, $\epsilon_{k+1}^{(i+3)}$, $\epsilon_{k+3}^{(i-1)}$ et $\epsilon_{k+3}^{(i+2)}$ en sont "éloignés".

On se donne une transformation anallagmatique A_1 appliquant ∞ en $u_1 \in R^m$ (par exemple une inversion de pôle u_1) et on définit un nouveau tableau ${}^1\epsilon_\ell^{(j)}$ dont les éléments encadrés dans le schéma ci-contre seront les transformés par A_1 des éléments homologués du tableau initial : les points ${}^1\epsilon_{k-1}^{(i+2)}$, ${}^1\epsilon_{k-1}^{(i+3)}$, ${}^1\epsilon_{k+1}^{(i)}$, ${}^1\epsilon_{k+1}^{(i+3)}$, ${}^1\epsilon_{k+3}^{(i-1)}$ et ${}^1\epsilon_{k+3}^{(i+2)}$ seront "éloignés" de u_1 tandis que ${}^1\epsilon_{k+1}^{(i+1)}$ et ${}^1\epsilon_{k+1}^{(i+2)}$ en seront "voisins".

Si nous posons ${}^1\epsilon_k^{(i+1)} = a_1 \in R^m$ (par exemple $a_1=0$), la règle du losange nous permet de calculer ${}^1\epsilon_k^{(i+2)}$, ${}^1\epsilon_k^{(i+3)}$, ${}^1\epsilon_{k+2}^{(i)}$ et ${}^1\epsilon_{k+2}^{(i+2)}$ qui seront "éloignés" de ∞, et ${}^1\epsilon_{k+2}^{(i+1)}$ qui en sera "voisin". On peut alors calculer ${}^1\epsilon_{k+3}^{(i)}$ et ${}^1\epsilon_{k+3}^{(i+1)}$ ("voisins" de u_1) puis ${}^1\epsilon_{k+4}^{(i-1)}$ et ${}^1\epsilon_{k+4}^{(i+1)}$ (éloignés de ∞), mais le calcul de ${}^1\epsilon_{k+4}^{(i)}$ est à proscrire (perte de précision par cancellation).

On se donne alors une seconde transformation anallagmatique A_2 appliquant ∞ en $u_2 \neq \infty$ (il est loisible de reprendre A_1) et on définit un 3e tableau ${}^2\epsilon_\ell^{(j)}$ dont les éléments encadrés dans le schéma ci-contre sont les transformés par A_2 des éléments homologues de ${}^1\epsilon_\ell^{(j)}$. Si nous posons ${}^2\epsilon_{k+1}^{(i+1)} = a_2 \neq \infty$, on peut alors calculer ${}^2\epsilon_{k+4}^{(i)}$ avec une bonne précision. Il suffit alors de prendre ${}^1\epsilon_{k+4}^{(i)} = A_2^{-1}({}^2\epsilon_{k+4}^{(i)})$ pour calculer ${}^1\epsilon_{k+5}^{(i-1)}$ et ${}^1\epsilon_{k+5}^{(i)}$ (non "voisins" de u_1), puis de revenir au tableau initial par $\epsilon_{k+3}^{(i)} = A_1^{-1}({}^1\epsilon_{k+3}^{(i)})$, $\epsilon_{k+3}^{(i+1)} = A_1^{-1}({}^1\epsilon_{k+3}^{(i+1)})$, $\epsilon_{k+5}^{(i-1)} = A_1^{-1}({}^1\epsilon_{k+5}^{(i-1)})$ et $\epsilon_{k+5}^{(i)} = A_1^{-1}({}^1\epsilon_{k+5}^{(i)})$.

On complètera alors le tableau initial en appliquant la règle du losange.

On notera que chacun des deux éléments $^1\varepsilon_{k+3}^{(i+1)}$ et $\varepsilon_{k+4}^{(i+1)}$ est calculé de 2 façons différentes, ce qui permet de contrôler la précision du résultat.

L'analyse de la progression des erreurs montre que ce schéma de calcul est nettement plus stable que les diverses variantes de l'ε-algorithme (scalaire ou vectoriel) proposées jusqu'à présent. Les essais numériques confirment cette analyse :

Si on applique l'ε-algorithme à la suite définie par :

$$x_0 = 4,\ x_1 = 3.6,\ x_2 = 3.226,\ x_3 = 2.22 \text{ et } x_n = x_{n-4}/2,\ \forall\ n \geq 4$$

on a $\varepsilon_8^{(n)} = 0,\ \forall\ n$. La mise en oeuvre de l'ε-algorithme classique et de la variante présentée ici fournit les valeurs suivantes de $\varepsilon_8^{(n)}$, selon que l'on travaille avec 4 ou 8 chiffres :

n	ε-algorithme classique		variante proposée	
	4 chiffres	8 chiffres	4 chiffres	8 chiffres
0		0.059 1782		0.000 0042
1		-0.063 5578		0.000 0036
2		0.020 8175	0.014	-0.000 0042
3	1.134	-0.016 5354	-0.0016	0.000 0012
4	0.6387	0.046 095	0.0051	0.000 0030
5	2.187	-0.051 4698	-0.0013	0.000 0032

Cet algorithme est naturellement plus coûteux que l'ε-algorithme classique. Son occupation mémoire est apparemment importante. On notera toutefois qu'on peut le mettre en oeuvre selon une technique de diagonale ascendante qui permet d'économiser notablement la place occupée.

REFERENCES :

[1] CORDELLIER F. : L'ε-algorithme vectoriel : interprétation géométrique et règles singulières, Colloque d'Analyse Numérique de Gourette (juin 1974).

[2] CORDELLIER F. : Particular rules for the vector ε-algorithm, Pub. 66, Labo. Calcul, Univ. Lille I (février 1976).

[3] CORDELLIER F. : Contrôle des erreurs dans les calculs vectoriels et application à l'ε-algorithme, Séminaire d'Analyse Numérique, Univ. Lille I (janvier 1976).

[4] HADAMARD J. : Récents progrès de la géométrie anallagmatique. Oeuvres (937-986).

[5] WYNN P. : Acceleration techniques for iterated vector and matrix problems, Math. Comp. 16 (1962) 301-322.

[6] WYNN P. : Singular Rules for certain non linear algorithms. BIT 3 (1963) 175-195.

Vers une mise en œuvre fiable et stable des algorithmes de losange

Résumé: Les algorithmes de losange consistent essentiellement en la construction récursive colonne par colonne d'un tableau triangulaire dont les deux premières colonnes sont définies par les données. Cette construction récursive repose sur une relation unique paramétrée par des coefficients dépendant de l'algorithme et des données. Nous avons déjà montré que, pourvu que ces coefficients soient liés par une certaine relation, ces algorithmes vérifient une propriété d'invariance homographique dans $\mathbb{C}$: la relation qui lie ceux de leurs éléments dont l'indice de colonne a la même parité reste vérifiée lorsqu'on leur applique globalement la même transformation homographique. Cette propriété est encore vraie dans $\mathbb{R}^n$ à condition de substituer les transformations anallagmatiques aux transformations homographiques, et cela nous avait permis de découvrir comment l'identité de Wynn s'étendait aux tables de Padé non normales et avait donc contribué à étendre notablement la fiabilité des algorithmes de mise en œuvre de l'ε-algorithme ou de calcul des approximants de Padé. Elle nous permet aujourd'hui d'améliorer de façon considérable la stabilité numérique de tous les algorithmes de losange qui la vérifient. On présente ici l'essentiel des idées qui conduisent à l'élaboration d'un algorithme exploitant cette opportunité et qu'on matérialise par un programme fortran. Quelques expériences numériques viennent illustrer l'intérêt de cet algorithme.

1 Introduction et motivation.

Les algorithmes de losange interviennent couramment en interpolation rationnelle: algorithme de Thiele [19] et de Claessens [4], variante [10] de l'algorithme de Stoer-Larkin [16,18] ou valeurs ponctuelles d'une table de Padé [22]. Ils interviennent encore dans la mise en œuvre de certaines transformations de suites: ρ et ε-algorithmes scalaires [2,21], vectoriels et leurs généralisations [3,23]. Ils consistent à construire une suite de suites (θ_i) à partir d'une initialisation fixant les deux premières suites et à calculer les termes des suites suivantes en utilisant une relation de récurrence:

Initialisation: $\theta_{-1}^j = v_j, \theta_0^j = x_j,\ j \in \mathbb{N}$.

Récurrence: $\theta_{i+1}^j = \theta_{i-1}^{j+1} + a_i^j / (\theta_i^{j+1} - \theta_i^j),\ i,j \in \mathbb{N}$.

où les coefficients a_i^j et les suites initiales (v_j) et (x_j) caractérisent l'algorithme.

Dans la notation θ_i^j, l'indice inférieur i identifie une suite tandis que l'indice supérieur j représente l'indice du terme dans la suite θ_i. Les éléments θ_i^j sont placés dans un tableau à deux indices où l'indice inférieur i repère une colonne et l'indice supérieur j une diagonale descendante (figure 1). Chaque relation de récurrence fait donc intervenir les éléments situés aux quatre sommets d'un losange et dépend d'un coefficient a_i^j attaché à ce losange (figure 2).

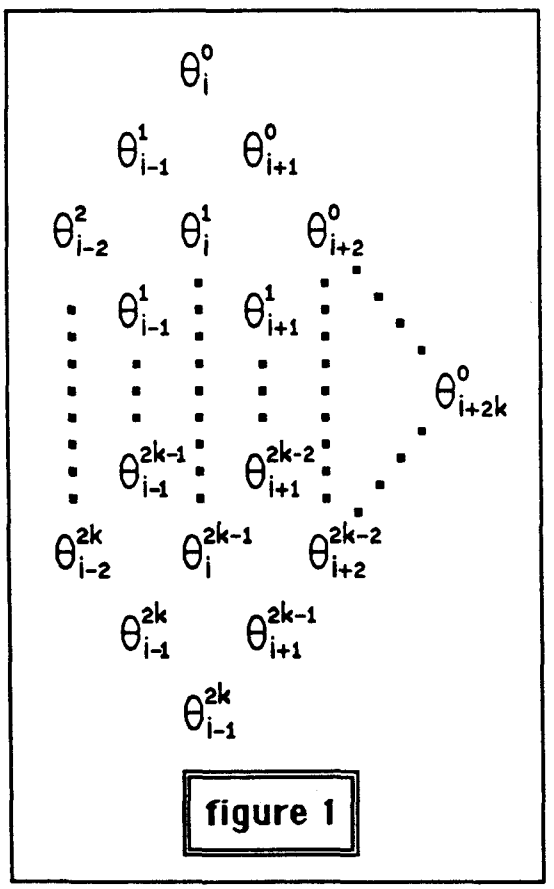


figure 1

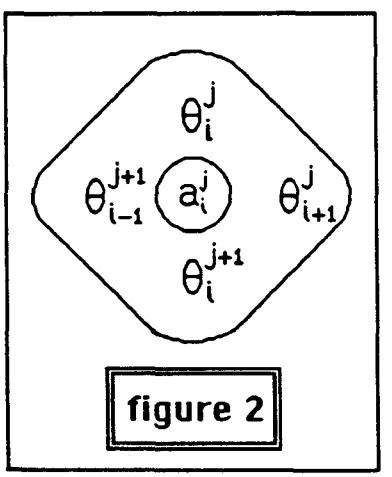


figure 2

La mise en œuvre de ces algorithmes est handicapée par deux difficultés:
1) leur fiabilité est doûteuse: l'égalité de deux éléments consécutifs d'une même colonne interdit le calcul de certains éléments de la colonne suivante;

2) leur stabilité numérique est médiocre: la trop grande proximité de deux éléments consécutifs d'une même colonne induit une grande erreur de calcul dans certains éléments de la colonne suivante.
Bien entendu, tout problème rencontré lors de la construction d'une colonne est hérité et même amplifié dans les colonnes suivantes. Nous nous proposons ici d'apporter une solution à ce double problème de la fiablité et de la stablité numérique.

2. Solutions apportées

Dès 1963, Wynn [25] a proposé deux règles pour lutter contre l'absence de fiabilité et l'instabilité numérique de l'ε-algorithme scalaire lorsque deux valeurs consécutives d'une même colonne sont égales (non fiabilité) ou voisines (instabilité). Brezinski [1] a présenté une règle similaire pour le ρ-algorithme scalaire, et j'ai également proposé une règle valable dans le cas de l'ε-algorithme vectoriel [6]. Toutefois ,le problème n'est pas règlé pour autant: que faire si plus de deux valeurs consécutive d'une même colonne sont égales ou même simplement voisines? L'extension des règles précédentes s'avère nécessaire.

La solution que je propose repose sur une notion fondamentale: l'invariance anallagmatique des colonnes de même parité de tout algorithme de losange dont les coefficients vérifient une certaine propriété. Esquissons ici brièvement les grandes lignes de la démarche qui aboutit aux résultats suivants:
1) extension de l'identité de Wynn [26] dans le cas vectoriel [9],d'où une version fiable (mais non nécessairement numériquement stable) de l'ε-algorithme vectoriel;
2) versions fiables et stables de tous les algorithmes de losange dont les coefficients vérifient une certaine propriété [5,10].

Sans hypothèse complémentaire, les éléments θ_i^j sont liés par l'identité suivante que nous appellerons *règle de la croix*, valable pour tout couple (i,j) d'entiers non négatifs:

$$a_i^j(\theta_i^j-\theta_i^{j+1})^{-1}+a_i^{j+1}(\theta_i^{j+2}+\theta_i^{j+1})^{-1}=a_{i+1}^j(\theta_{i+2}^j-\theta_i^{j+1})^{-1}+a_{i-1}^{j+1}(\theta_{i-2}^{j+2}-\theta_i^{j+1})^{-1}$$

A tout algorithme de losange, on peut donc associer deux algorithmes qui ne font intervenir que les colonnes dont l'indice a la même parité, chacun étant initialisé au moyen de deux colonnes: les colonnes d'indices -2 et 0 pour l'un, celles d'indices -1 et +1 pour l'autre. Nous nous réfèrerons à ces algorithmes comme à la forme en croix (paire ou impaire) d'un algorithme de losange. Notons encore que, si l'on connait toutes les colonnes dont l'indice est pair (respectivement impair), il suffit d'une valeur supplémentaire dans l'une quelconque des autres colonnes pour définir les valeurs des colonnes ayant l'autre parité.

Rappelons qu'on qualifie d'anallagmatique toute transformation ponctuelle qui conserve la nature des cercles et des sphères (en considérant les hyperplans comme des hypersphères contenant le point à l'infini) et le birapport de quatre points sur un cercle. Ces transformations constituent le sous-groupe des transformations ponctuelles de l'espace engendré par les similitudes et les inversions [13]. Toute la suite repose sur la remarque suivante:
Pourvu que les poids a_i^j satisfassent la relation:

$$a_i^j+a_i^{j+1}=a_{i-1}^{j+1}+a_{i+1}^j,\quad \forall\ i,j\in\mathbb{N},$$

la règle de la croix est anallagmatiquement invariante [10].
Nous nous restreindrons désormais aux algorithmes de losange dont les poids satisfont cette propriété.

Appliquons une transformation anallagmatique A à deux colonnes consécutives de même parité d'un tableau T construit lors de la mise en œuvre d'un algorithme de losange. A partir de l'initialisation définie par les transformées par A de ces deux colonnes, mettons en œuvre la forme en croix de l'algorithme de losange utilisant les mêmes coefficients que ceux utilisés lors de la construction de T. Alors les colonnes de même parité que l'initialisation construites au moyen de cette technique sont les transformées par A des colonnes de même indice du tableau initial (figure 3).

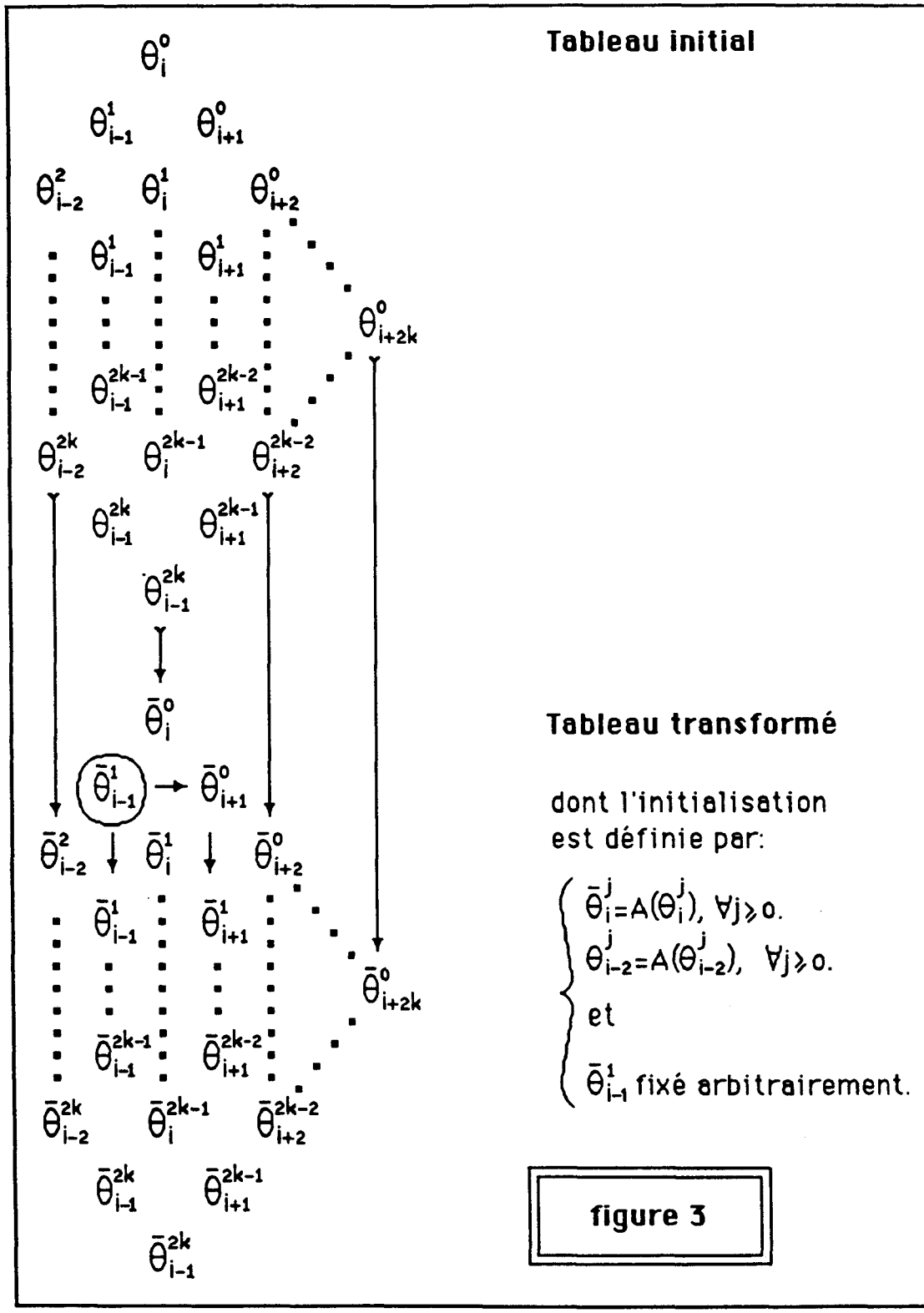


figure 3

Dans toute la suite, nous n'utiliserons qu'une seule transformation anallagmatique: l'inversion ayant pour pôle l'origine et pour puissance l'unité qui offre l'avantage d'être numériquement la plus stable de ces transformations. La propriété d'invariance anallagmatique peut alors être utilisée de deux façons distinctes que nous qualifierons respectivement de systématique et sélective.

2.2 Utilisation systématique

Notons $^{0}\theta_i^j$ les éléments calculés au moyen de l'algorithme de losange initial. Dès que la colonne d'indice 1 est connue, on peut inverser les éléments des deux colonnes d'indices respectifs -1 et 1 qu'on retient comme initialisation d'un nouvel algorithme de losange utilisant les mêmes coefficients et dont les éléments seront notés $^{1}\theta_i^j$. Pour assurer une mise en œuvre commode de cet algorithme, on fixera arbitrairement une valeur dans les colonnes d'indice pair, par exemple $^{1}\theta_0^1 = 0$, ce qui permet de calculer toutes les colonnes du nouveau tableau.
On itère le procédé en inversant les éléments des colonnes d'indices 0 et 2 du tableau $^{1}\theta_i^j$ pour des éléments qu'on choisit comme initialisation d'un nouvel algorithme de losange utilisant encore les mêmes coefficients et dont les éléments seront notés $^{2}\theta_i^j$. On introduira des colonnes impaires dans ce tableau en fixant arbitrairement une valeur, par exemple $^{2}\theta_i^0 = 0$. On pourra ainsi définir des tableaux successifs $^{(0)}T$, $^{(1)}T$, $^{(2)}T$, ... , $^{(k)}T$, ... , dont les éléments sont notés $^{0}\theta_i^j$, $^{1}\theta_i^j$, $^{2}\theta_i^j$, ... , $^{k}\theta_i^j$, ...

Les éléments homologues $^{m-1}\theta_i^j$ et $^{m}\theta_i^j$ de deux tableaux consécutifs $^{(m-1)}T$ et $^{(m)}T$ sont inverses l'une de l'autre s'ils appartiennent à des colonnes de même parité, donc si les indices m et i ont même parité. Si nous disposons d'un moyen de connaître la précision avec laquelle est connu un élément calculé, on peut alors comparer les éléments homologues de deux tableaux consécutifs, retenir celui qui est le plus précis et recalculer l'autre par inversion. Cela reste valable si la précision d'un des deux nombres est nulle, c'est à dire si ce nombre est indéterminé, ce qui permet d'améliorer la fiabilité.
L'utilisation systématique de cette idée conduit à la version fiable et stable de divers algorithmes de losange qui est proposée ici, version dans laquelle l'estimation de l'erreur est fournie par la méthode de perturbation aléatoire de La Porte et Vignes [15,20]. Bien entendu, pour des raisons d'économie, on peut limiter le nombre des tableaux créés, donc le coût de l'algorithme.

2.3 Utilisation sélective.

On peut aussi circonscrire la mise en œuvre de cette technique d'inversion exploitant l'invariance anallagmatique d'un algorithme de losange à des zônes bien délimitées du tableau construit, zônes dans lesquelles on aura détecté l'instabilité numérique ou même la non fiabilité de l'algorithme initial. Cette méthode sélective est simplifiée si l'on dispose d'informations précises sur la structure des singularités, ce qui est précisément le cas le l'ε-algorithme (scalaire ou vectoriel). On peut alors l'exploiter dans deux contextes distincts:

1) Etablissement de règles singulières généralisées pour l'ε-algorithme vectoriel, ce qui conduit à deux résultats [7]: d'une part, une version fiable mais instable de l'ε-algorithme et d'autre part, une extension de l'identité de Wynn pour la table de Padé scalaire. Cette identité de Wynn généralisée a été récemment étendue à des objets qui généralisent la notion d'approximant de Padé: les "multivariate approximants" par A. Cuyt [11] et les "GIPAs" par P. Graves-Morris et Jenkins [12].

2) Proposition d'une version fiable et numériquement stable de l'ε-algorithme vectoriel. Présenté au colloque de Port-Bail en 1976 [5], cet algorithme a été programmé par R.Khéloufi [14] en 1985.

Le fait qu'on sache que, dans le cas de l'ε-algorithme, les singularités ou les quasi-singularités sont circonscrites dans des blos carrés simplifie cette utilisation sélective. La détection et la localisation précise des blocs quasi-singuliers reste encore une question très délicate. Dans le cas des autres algorithmes de losange, l'utilisation sélective est encore plus délicate, voire dangereuse. On lui préfèrera alors une utilisation systématique plus coûteuse certes, mais beaucoup plus fiable.

2 Aspects arithmétiques.

2.1 Introduction.

Le choix de l'arithmétique utilisée est l'un des aspects essentiels de ce travail. Ce choix sera guidé par deux motivations:

1° On contribuera à la fiabilité de l'algorithme et à la souplesse de la programmation en introduisant deux valeurs spéciales dans l'arithmétique utilisée: l'infini et l'indéfini.

2° L'algorithme proposé repose essentiellement sur la notion de choix: très fréquemment se posera la question de savoir quelle approximation d'une valeur numérique on peut retenir pour la suite du calcul. Cela suppose qu'on sache apprécier la qualité d'une telle approximation.

La première nous conduit donc à proposer un type arithmétique de base qui comporte les deux éléments infini et indéfini tandis que la seconde nous amène à envisager un type arithmétique comportant deux informations de base: la valeur numérique et une information sur la précision avec laquelle cette valeur numérique est connue. Le modèle arithmétique retenu satisfera ces deux exigences.

2.2 L'arithmétique compactifiée.

2.1.1 L'arithmétique compactifiée théorique.

Les symboles $\mathbb{R}$ et $\bar{\mathbb{R}}$ représentent respectivement la droite numérique simple et compactifiée par l'adjonction d'un point à l'infini unique noté ∞ avec les conventions habituelles:

$x\in\mathbb{R} \Rightarrow x+\infty = \infty+x = x-\infty = \infty-x = \infty/x = \infty$ et $x/\infty=0$.

$x\in\bar{\mathbb{R}} \Rightarrow x\times\infty = \infty\times x = x/0 = \infty$ et $0/x=0$.

Pour éviter que certaines opérations internes ne soient pas définies, ajoutons un élément unique appelé indéfini et noté ω pour former: $\mathbb{R}' = \mathbb{R}\cup\{\omega\}$ et $\bar{\mathbb{R}}' = \bar{\mathbb{R}}\cup\{\omega\} = \mathbb{R}'\cup\{\infty\}$.

On complètera la définition des opération élémentaires par:

$\infty+\infty = \infty-\infty = 0\times\infty = \infty\times 0 = 0/0 = \infty/\infty = \omega$.

Enfin, toute opération dont un opérande est indéfini a l'indéfini pour résultat.

2.2.2 L'arithmétique machine classique.

La grande majorité des arithmétiques utilisées en calcul automatique peuvent être modélisés de la façon suivante: on se donne un sous-ensemble fini S de $\mathbb{R}$ comportant 0 et tel que: $x\in S \Rightarrow -x\in S$. Puisque cet ensemble est fini, le sous-ensemble des nombres positifs de S comporte une borne inférieure $m = \min\{x\in S \mid x>0\}$ et une borne supérieure $M=\max\{x\in S\}$, qu'on appelle respectivement seuils inférieur et supérieur de dépassement de capacité. On se donne une projection Π_S de [-M,M] sur S respectant la relation d'ordre sur $\mathbb{R}$: $x<y \Rightarrow P(x)\leq P(y)$.

Les arithmétiques flottantes courantes sont telles qu'il existe un nombre τ (appelé souvent précision de la représentation ou encore zéro-machine) tel que: $|x|\in[m,M]\Rightarrow|P(x)/x-1|\leq\tau$, mais cette propriété pourtant essentielle ne sera pas explicitement exploitée dans ce travail.

A toute opération binaire interne définie sur $\mathbb{R}$ et notée * (application de $\mathbb{R}\times\mathbb{R}$ dans $\mathbb{R}$ avec la restriction de la division par 0), on associe une opération machine notée $(*)_s$ par: $a(*)_s b = \Pi_S(a*b)$.

Nous supposerons que tout est mis en œuvre pour que les opérations arithmétiques de base respectent ce modèle (présence de chiffres de garde en particulier).

2.2.3 Spécification d'une arithmétique compactifiée effective.

Nous voulons décrire ici une arithmétique comportant certains des avantages de l'arithmétique compactifiée tout en respectant les contraintes qu'impose la représentation des nombres en machine. Nous retiendrons le modèle suivant:
on se donne un sous-ensemble fini S_1 de $\mathbb{R}\setminus\{0\}$ tel que: $s\in S_1 \Rightarrow -s\in S_1$, auquel sont associés les deux nombres $m_1=\min\{s\in S_1 \mid s>0\}$ et $M_1=\max\{s\in S_1\}$; on lui adjoint trois nouveaux éléments qu'on identifie aux éléments correspondants de $\bar{\mathbb{R}}$': zéro (noté 0), l'infini (noté ∞) et l'indéfini (noté ω) pour former un ensemble noté S_1'': $S_1''=S_1\cup\{0,\infty,\omega\}$. On se donne une projection de $\bar{\mathbb{R}}$' dans S_1'' satisfaisant les contraintes suivantes:

$$1^\circ\ \Pi_{S_1''}(x)=\begin{cases}\omega & \text{si } x=\omega\\ 0 & \text{si } |x|<m_1\\ \infty & \text{si } |x|>m_1\\ s\in S_1'' & \text{sinon}\end{cases}$$

$$2^\circ\ -M\leq x\leq y\leq M \Rightarrow \Pi_{S_1''}(x)\leq\Pi_{S_1''}(y).$$

A toute opération binaire interne * définie sur $\bar{\mathbb{R}}$', on associe une opération image notée $(*)_{S_1''}$ définie sur S_1'' par: $x\,(*)_{S_1''}\,y = \Pi_{S_1''}(x*y)$, $\forall x,y\in S_1''$.

Remarques:
Certaines opérations auront donc pour résultat la valeur infinie ou même la valeur indéfinie, mais le fait que ces valeurs soient intégrées à l'arithmétique autorisera la poursuite de l'algorithme.

D'autre part, la relation d'ordre sur $\mathbb{R}$ ne s'étendant ni à $\bar{\mathbb{R}}$, ni à $\mathbb{R}$, les valeurs ∞ et ω ne pourront jamais figurer dans un autre test que l'égalité ou la non égalité.

2.2.4 Implémentation d'une arithmétique compactifiée.

Pour représenter une telle arithmétique, il est commode d'exploiter l'arithmétique machine de base disponible sur un matériel donné. Il est aisé de proposer: m_1 sera la plus petite puissance entière de la base dont le carré reste strictement supérieur à m, tandis que M_1 sera la plus grande puissance entière de la base dont le carré reste strictement inférieure à M. Cette restriction offre l'avantage d'autoriser l'utilisation de l'arithmétique machine sans craindre les dépassements de capacité. D'où le choix: $S_1=\{x\in S \mid m_1\leq|x|\leq M_1\}$.

Conformément à ce qui a été dit plus haut, toute valeur extérieure à cet intervalle sera forcée à zéro si elle est trop petite, ou à l'infini si elle est trop grande. Pour achever la description de cette représentation, il faut encore préciser la représentation qui a été retenue pour les trois valeurs particulières 0,∞ et ω, représentation que nous noterons r(0),r(∞) et r(ω). Cette représentation doit satisfaire les contraintes suivantes:

1°) les trois représentations ne doivent pas appartenir au domaine de calcul courant:

$r(\alpha) \notin [m_1, M_1]$, pour $\alpha \in \{0, \infty, \omega\}$.

2°) les représentations de zéro et l'infini sont inverses l'une de l'autre.

En fait, j'ai retenu: $r(0)=(m_1)^2$, $r(\infty)=(M_1)^2 (=1/r(0))$ et $r(\omega)=-r(\infty)$, mais ces choix sont bien arbitraires.

2.3 Estimation de l'erreur.

2.3.1 Divers modes d'estimation.

L'estimation de l'erreur peut revêtir deux formes fondamentales:

1°) être une majoration fiable mais souvent pessimiste de l'erreur effective qui entache la représentation d'un nombre;

2°) fournir une estimation de l'ordre de grandeur probable de cette erreur, mais cette estimation peut parfois être sous-évaluée.

Qu'il s'agisse d'une majoration ou d'une estimation, l'outil de mesure peut revêtir lui aussi deux formes distinctes:

1°) il peut être décrit par le programmeur qui, en plus de la manipulation de ses nombres, doit encore gérer celle des erreurs qui les entachent;

2°) il peut encore être intégré à l'arithmétique de base au moyen de laquelle seront programmés les calculs, que ce soit fait au niveau du software (notion de paquetage en ADA, qu'on simulera avec des sous-programmes en fortran) ou que cela soit fait au niveau du hardware.

La solution que nous avons retenue ici est une estimation de l'erreur de calcul obtenue par une technique de simulation intégrée à l'arithmétique que l'on doit à Vignes et Laporte [13], la méthode des perturbations aléatoires.

2.3.2 Idée de la méthode de perturbation.

Supposons qu'on ait mis en œuvre un algorithme A avec l'arithmétique flottante classique d'un ordinateur caractérisée par un domaine de calcul S et une projection P_S pour obtenir un résultat r (éventuellement un n-uple). Analysons l'exécution de cet algorithme à partir de ses données. Chaque résultat intermédiaire étant la projection (selon P_S) du résultat effectif de l'opération, une erreur portant sur le dernier digit de la représentation est susceptible d'entacher ce résultat. L'erreur résultante qui affecte le résultat final r est la conséquence de toutes ces approximations intermédiaires. Si nous supposons que toutes les opérations sont différentiables, l'erreur entachant le résultat final peut s'écrire sous la forme: $\delta r = \sum_{\lambda \in \Lambda} \frac{\partial r}{\partial x_\lambda} \times \varepsilon_\lambda$, où Λ repère l'ensemble des résultats intermédiaires, ε_λ représente l'erreur absolue qui entache le résultat intermédiaire x_λ et $\frac{\partial r}{\partial x_\lambda}$ est la dérivée partielle de r par rapport à ce même résultat x_λ. L'idée de la méthode de perturbation consiste à introduire des erreurs ε'_λ qui soient du même ordre de grandeur que les erreurs effectives ε_λ inconnues et de constater l'effet global de ces erreurs sur le résultat r.

2.3.3 Représentation effective. Modes de simulation.

a) Mode bruité.

Pour obtenir un perturbation ε'_λ du même ordre de grandeur que l'erreur effective ε_λ, on modifie lors de chaque opération le dernier digit de la mantisse d'une unité (on ne peut pas moins). Dans le cas où la base est une puissance entière de deux, cela est réalisé en changeant le dernier chiffre binaire du codage. Cette méthode, appelée mode bruité [17] est d'un usage limité car elle ne permet qu'une seule perturbation, donc une seule estimation de l'erreur qui entache le résultat final.

b) perturbations séquentielles.

Pour éviter à cette technique le caractère figé que nous venons de dénoncer, Vignes et Laporte [15] proposent la méthode suivante: on va exécuter plusieurs fois le même calcul et, pour chaque opération, on tirera au sort pour savoir si on utilise l'arithmétique courante ou le mode bruité. Ainsi peut-on espérer que la moyenne des erreurs introduites lors des diverses exécutions sera une représentation plus réaliste de l'erreur effectivement introduite dans le calcul en mode courant. On dispose alors d'autant d'approximations du résultat final qu'il y a d'exécutions différentes. Si la connaissance de la précision avec laquelle est calculée le seul résultat final nous suffit, on pourra donc reprendre le calcul plusieurs fois de suite, le caractère aléatoire des diverses exécutions nous assurant une bonne répartition des résutats: c'est la dispersion des résultats qui nous fournit l'estimation de l'erreur que nous souhaitons. De plus, la cohérence des résultats pourra être apprécíée lors des diverses exécutions en mode bruité sélectif: le nombre d'exécutions distinctes, minimal si les diverses estimations de l'erreur sur le résultat sont cohérentes, pourra être éventuellement augmenté dans le cas (rare) où les premières exécutions en mode bruité sélectif fournissent des indications contradictoires quant à l'estimation de la précision du résultat final.

c) perturbations simultanées.

Par contre, si nous souhaitons, comme c'est ici notre cas, avoir des informations actualisées sur la précision des résultats intermédiaires, il nous faut alors une exécution simultanée de toutes ces calculs perturbés. Cela n'est possible que si nous fixons a priori le nombre le nombre des simulations et, de ce point de vue, cette méthode est moins souple que la précédente. A cette fin, représentons chaque nombre par un vecteur à p composantes. Chaque opération binaire de l'algorithme sera mise en œuvre simutanément de p-1 façons: pour i=1,...,p-1, on prendra la i-ième composante de chacun des deux vecteurs-opérandes, on les composera avec l'opération binaire en question, et le résultat sera rangé dans la i-iéme composante du vecteur-résultat. Pour i=1, le calcul sera réalisé selon les règles de l'arithmétique dont on dispose sur le matériel de calcul utilisé, mais pour les p-2 autres, le résultat sera éventuellement perturbé avant d'être rangé, la perturbation consistant à changer la parité du dernier bit en respectant une procédure aléatoire. Notons toutefois que l'effet d'une erreur ne se fera sentir que dans le seul cas où au moins l'une des perturbations qui affectent les p-2 composantes est effective: puisque les p-2 perturbatoins sont aléatoires et indépendantes, il se peut qu'aucune ne le soit. Dans ce cas, l'estimation de l'erreur introduite dans cette opération est nulle, et la contribution de cette opération élémentaire dans l'estimation de l'erreur qui va entacher les résultats ultérieurs est également nulle. Si cette opération est précisément celle qui contribue le plus à l'incertitude du résultat final, l'estimation de l'erreur risque d'être totalement fausse. Pour éviter ce phénomène pervers que la mise en œuvre séquentielle n'avait pas révélé, mais qui n'était pas absent pour autant, on introduira une erreur systématique dans une des composantes à perturber aléatoirement. Cela signifie que l'une des p-2 composantes d'indice i=2,...,p-1 peut ne pas être perturbée aléatoirement mais de façon systématique.

Deux solutions s'offrent à nous:
1°) mettre en œuvre la méthode de perturbation aléatoire sur les p-2 composantes, tester si l'une de ces perturbations est effective, et perturber une de ces composantes si ce n'est pas le cas.
2°) perturber systématiquement une des composantes, les autres étant perturbées de façon aléatoire.

Dans chacune de ces deux procédures, il est question de perturber une des composantes. Le choix de la composante à perturber peut être défini de trois façons distinctes:
a) la composante est fixée une fois pour toutes (c'est alors appliquer le mode bruité à cette composante);
b) faire appel à une procédure garantissant un balayage de l'ensemble des composantes (balayage cyclique par exemple);
c) tirer au sort la composante au moyen d'une procédure aléatoire.
La méthode (2b) que nous avons retenue diffère légèrement de celle que préconise Vignes (1a) dans la méthode CESTAC [20].

Comme dans l'utilisation séquentielle, l'erreur sera estimée à partir des écarts entre les p-1 approximations d'un nombre contenues dans un vecteur, mais ici, l'erreur qui affecte tout résultat intermédiaire est accessible dès que ce résultat a été calculé: on peut en effet retenir la moyenne des écarts, voire le plus grand écart comme estimation de l'erreur dont est entachée la première composante de chaque nombre-vecteur. Cette représentation de l'erreur peut indifféremment être calculée quand on en a besoin ou en même temps que le résultat effectif de l'opération: c'est cette dernière solution qui a été retenue; c'est pourquoi on trouvera cette estimation de l'erreur dans la p-ième composante de chaque nombre-vecteur calculé.

2.4 Estimation riemanienne de l'erreur.

2.4.1 Nécessité d'un type d'erreur spécifique.

La représentation des erreurs numériques est moins triviale qu'il ne paraît. Si l'on préfère le plus souvent adopter les erreurs relatives bien adaptées à la représentation flottante des nombres, on sait que cette représentation sera en défaut dans le cas d'erreurs entachant un nombre nul. Le fait d'adjoindre les valeurs ∞ et ω pour compléter l'arithmétique classique n'a pas simplifié notre tâche. Certes, il est clair que l'erreur qui entache ω est surement plus grande que celle qui entache tout autre nombre: aussi n'aurons nous guère de peine à la définir. Il n'en est pas de même pour les deux autres valeurs spéciales que sont le zéro et l'infini. Nous venons de rappeler que la notion d'erreur relative est à rejeter dans le cas de la représentation de zéro, mais elle est encore inutilisable dans la représentation de l'infini. De même, si la notion d'erreur absolue reste applicable au voisinage de zéro, elle devient également impraticable dans la représentation de l'infini. Pourtant, tout numéricien reconnait que, dans la plupart des cas, 10^{-8} est une approximation convenable de zéro, moins bonne que 10^{-12} soit, mais meilleure que 10^{-4} tandis que 10^{8} sera une approximation tout aussi acceptable de l'infini, moins bonne que 10^{12} mais meilleure que 10^{4}. Il serait souhaitable de disposer d'une représentation de l'erreur qui respecte ce point de vue.

2.4.2 La distance riemanienne. [8,13]

Dans le plan complexe, notons ϕ l'inversion de puissance un et dont le pôle a l'affixe i. Cette inversion transforme la droite réelle $\mathbb{R}$ en le cercle C de diamètre unité centré au point d'affixe i/2 (voir figure 4).

Compléter $\mathbb{R}$ en $\bar{\mathbb{R}} = \mathbb{R} \cup \{\infty\}$ équivaut à munir la droite réelle de la topologie image par $\phi^{-1}=\phi$ de la topologie induite sur le cercle C par la topologie naturelle de $\mathbb{C}$. Cette topologie peut dériver de deux distances élémentaires:
1°la distance attachée au module de la différence de deux complexes:

$d: x,y \in \mathbb{R} \rightarrow d(x,y) = |\phi(x)-\phi(y)| \in [0,1]$

2°la distance angulaire des deux images par ϕ:

$d': x,y \in \mathbb{R} \rightarrow d'(x,y) = |Arg(\phi(x)-i/2)-Arg(\phi(y)-i/2)| \in [0,\pi]$.

Nous retiendrons la première de ces distances pour laquelle un calcul élémentaire donne:

$$d(x,y)=\begin{cases}\dfrac{|x-y|}{(1+x^2)^{1/2}(1+y^2)^{1/2}} & \text{si } x \text{ et } y \in \mathbb{R} \\ (1+x^2)^{-1/2} & \text{si } x \neq y=\infty \\ (1+y^2)^{-1/2} & \text{si } y \neq x=\infty \\ 0 & \text{si } x=y=\infty\ .\end{cases}$$

La distance de deux nombres est maximale si l'un est l'opposé de l'inverse de l'autre. C'est en particulier le cas de zéro et l'infini, ou de l'unité et de son opposé. Cette distance est alors égale à l'unité. On notera que, pour tout couple x,y de $\bar{\mathbb{R}}$, on a: $d(x,y) = d(1/x,1/y)$. En particulier, pour $y=\infty$ on a: $d(x,\infty) = d(1/x,0)$.

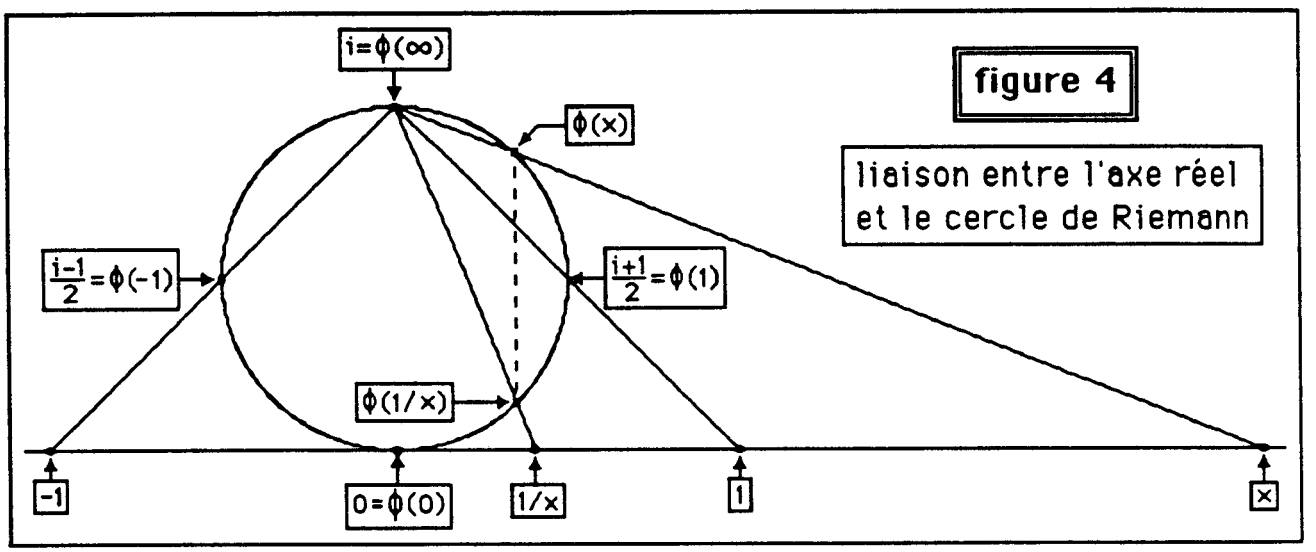


figure 4

liaison entre l'axe réel et le cercle de Riemann

D'autre part, on se fera une meilleure idée de cette distance en notant qu'on a les approximations suivantes:

$d(x,y)$ est voisin de $\dfrac{|x-y|}{2}$ si x et y sont proches de l'unité, voisin de $|x-y|$ si x et y sont proches de zéro et voisin de $\left|\dfrac{1}{x}-\dfrac{1}{y}\right|$ si x et y sont proches de l'infini.

Ainsi la distance de 10^{12} à -10^{16} est-elle voisine de celle de 10^{-12} à -10^{-16}, donc proche de 10^{-12}. Cette notion assez déroutante ne doit pas être maniée sans précaution car elle conduit à considérer comme très proches des nombres très grands n'ayant aucun chiffre commun et pouvant même être de signes contraires.

2.4.3 Représentation de l'erreur.

Convenons d'attacher à chaque nombre la plus grande distance riemanienne entre la composante principale (correspondant à l'indice i=1) et ses p-2 composantes secondaires (correspondant aux calculs en mode perturbé). Cette distance sera calculée après chaque opération dont le résultat est un nombre-vecteur et affectée à la p-ième composante de ce nombre-vecteur: $x_p = \max_{i=2..p} d(x_i, x_1)$.

Tout nombre-vecteur dont une composante est indéfinie est indéfini; l'erreur qui l'affecte est automatiquement fixée à 1, la valeur maximale des distances riemaniennes. Réciproquement, si un nombre-vecteur est affecté d'une erreur voisine de l'unité, alors ce nombre est presque indéfini: la notion d'erreur riemanienne permet ainsi de quantifier la notion de définition d'un nombre.

Il est souvent possible de disposer de plusieurs approximations d'un même résultat numérique. L'algorithme que nous proposons ici exploite justement le fait que la majeure partie des résultats intermédaires peuvent être calculés de deux façons distinctes. Pour que l'algorithme fonctionne dans de bonnes conditions, il faut encore que les estimations d'erreurs dont nous disposons soient correctes. S'il n'est guère aisé de contrôler cette correction, un contrôle plus élémentaire reste à notre disposition. A cette fin, nous dirons que deux approximations d'un même nombre sont compatibles si la distance de ces deux approximations est inférieure à la somme des erreurs entachant ces approximations (inégalité triangulaire). La vérification de la compatibilité de deux approximations d'un même nombre est un outil permettant de vérifier indirectement la validité des estimations de l'erreur: en effet, si deux approximations distinctes d'un même nombre sont incompatibles, cela implique que l'estimation d'erreur de l'un des deux nombres est incorrecte.

2.4.4 Définition des opérations.

Pour définir les opérations, nous avons tenu compte du fait que certaines opérations courantes ont un résultat indéfini: $\infty+\infty, \infty-\infty, 0\times\infty$ et $\infty\times 0$. Toutefois, le modèle retenu n'est pas totalement exempt de toute aberration. C'est ainsi que si le nombre a est très grand mais distinct de ∞, la somme $a+\infty$ donne ∞, alors que, pour $a=\infty$, ce résultat est ω. Pour éviter ce phénomène, on forcera à l'infini tout valeur voisine de l'infini, comme on forcera à zéro toute valeur voisine de zéro. Par voisine, nous entendrons ici: $d(a,b)\leq 2\times a_p$ où $b=0$ ou ∞. Cet artifice a permis de réduire notablement le nombre des approximations incohérentes rencontrées lors de la mise en œuvre de l'algorithme.

3. Algorithme de principe

3.1 Caractérisation du problème.

3.1.1 Les données.

Définissons les deux ensembles d'indices suivants:

$$\left.\begin{aligned}\Pi_n^l &= \{(k,j)\in \mathbb{Z}\times\mathbb{N} \mid 0\leq j+l\leq n \ ; \ l\leq k+j\leq n \ ; \ k+2\geq l\}\\ \Lambda_n^l &= \{(k,j)\in \mathbb{Z}\times\mathbb{N} \mid 0\leq j+l< n; \ l\leq k+j< n; \ k+2> l\}\end{aligned}\right\} \text{ où } l\in[0,p] \text{ et } p<n.$$

Ces ensembles d'indices permettent de repérer les points d'un quadrillage en quinconce de la forme de ceux qui sont représentés dans la figure 5.

Pour $(k,j)\in\Lambda_n^0$, nous supposons donnés des nombres $a_k^{(j)}$ vérifiant les relations suivantes:

$$(A) \qquad a_{k-1}^{(j-1)}+a_{k+1}^{(j)}=a_k^{(j)}+a_k^{(j+1)}, \text{ pour } 0\leq k,j,k+j\leq n-2.$$

Nous supposons donnée une suite finie de nombres $(x_j)_{0\leq j\leq n}$ et un nombre fixe $\lambda\in\bar{\mathbb{R}}$.

3.1.2 Le tableau ${}^l\theta_k^j$.

Considérons un tableau ${}^l\theta_k^j$ dont les trois indices vérifient $(k,j)\in\Pi_n^l$, $l=0,\ldots,p$, p étant, tout comme n, un entier non négatif donné. Ce tableau sera soumis à trois sortes de contraintes:

1° Certaines de ses valeurs sont fixées:

$^{0}\theta^{j}_{-2}=\lambda,\ j=2,\ldots,n;$

$^{0}\theta^{j}_{0}=x_j,\ j=0,\ldots,n;$

$^{1}\theta^{1}_{l-1}=0,\ l=2,\ldots,p.$

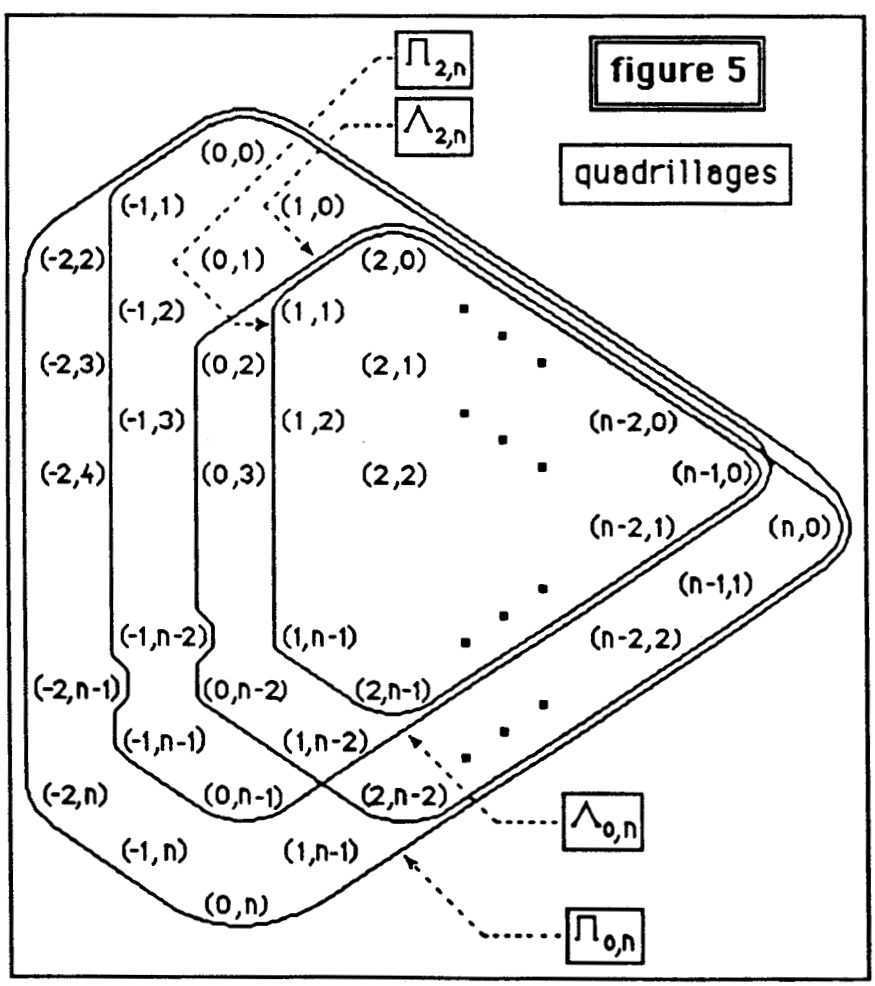
2° Pour chaque indice $l\in 0{:}p$, à chaque couple $(k,j)\in\Lambda_l$, est attachée une relation qui lie les valeurs des quatre sommets d'un losange:

$$^{l}\theta^{j}_{k+1}.-{}^{l}\theta^{j+1}_{k-1}.=a^{(l)}_{k}/({}^{l}\theta^{j+1}_{k}.-{}^{l}\theta^{j}_{k}.)$$

3° Pour $l\in 1{:}p$, à chaque couple $(k,j)\in L_l$ tel que $k+l$ soit impair, est attachée la relation suivante:

$$^{l}\theta^{j}_{k}=H_l({}^{l-1}\theta^{j}_{k}),$$

où H_l est une transformation homographique quelconque. Pour simplifier, nous supposerons que cette transformation homographique est de la forme: $H_l(x)=1/(x-\alpha)$, et sauf indication contraire, nous choisirons: $\alpha=0$.



3.1.3 Le problème posé.

Etant donnés deux entiers p<n, une suite finie de réels $(x_i)_{i=0..n}$ et un tableau de réels $(a_k^{(j)})$ définis pour $(k,j)\in \Lambda_n^0$ et satisfaisant la condition d'additivité (A), on se propose de déterminer le tableau ${}^l\theta_k^j$ qui satisfera les contraintes attachées à ces données.

Remarque: pour p=0, c'est la caractérisation des relations qui régissent la mise en œuvre de l'agorithme de losange classique attaché à ces données. Les autre relations sont des contraintes traduisant l'invariance homographique des colonnes de même parité.

3.2 Un algorithme de principe coûteux.

3.2.1 Le contexte.

Pour résoudre le problème posé, on suppose qu'on dispose d'une représentation des nombres comportant deux informations: la valeur approchée de ce nombre, et la précision avec laquelle cette valeur connue approche la valeur exacte. Chaque opération élémentaire nous fournit donc deux résultats: la valeur effective et la précision de cette valeur. Nous proposons tout d'abord un algorithme théorique dont le seul intérêt est de donner une idée synthétique de la stratégie que nous développerons dans le chapitre suivant.

3.2.2 Initialisation de l'algorithme.

a) initialisation générale par défaut:

pour $l=0,\ldots,p$ faire
  pour $(k,j)\in \Lambda_n^l$ faire
    ${}^l\theta_k^j := \omega$;

b) initialisation effective:

pour j:=0 à n faire ${}^0\theta_0^j = x_j$;

pour j:=2 à n faire ${}^0\theta_{-2}^j = \lambda$;

pour l:=2 à p faire ${}^l\theta_{l-1}^l = 0$;

3.2.3 boucle principale

tant que (amélioration (θ,l,k,j,val)) faire ${}^l\theta_k^j = val$;

où la fonction booléenne amélioration prend la valeur vrai dès que l'une des relations (losange ou inversion) permet d'affecter à une composante particulière du tableau θ une valeur plus précise que celle qui était connue lors de son appel. Les indices identifiant la valeur modifiée sont affectés aux paramètres l,k,j et la nouvelle valeur au paramètre val, le tableau θ étant inchangé. L'algorithme s'arrête dès qu'aucune des relations (de losange ou d'inversion) ne permet d'améliorer la précision d'un seul élément du tableau. Toutefois, nous ne tenterons pas de montrer que cette éventualité se présente nécessairement: la fiabilité de cet algorithme théorique (que nous n'envisageons pas de programmer) n'est pas établie.

3.2.4 Description de l'amélioration.

On définit un ordre privilégié dans lequel on contrôlera les relations. Le choix de cet ordre n'est pas essentiel. Dans cet algorithme de principe, nous traiterons d'abord l'ensemble des losanges, puis l'ensemble des inversions, mais tout autre ordre est permis, le balayage des losanges et des couples de valeurs inverses pouvant d'ailleurs être entrelacés. Par contre, il est essentiel qu'aucune relation ne soit omise.

```
fonction amélioration (θ: in tableau; i,j,k: ex indices; val: ex réel): booléen;
trouvé:booléen:=faux; l,k1,j1: indices;

debut

--1°recherche d'un losange non saturé:

pour l:=0 à p faire
     pour (k1,j1)∈ Λ^l_n faire

     s'il n'y a pas plus d'un sommet indéfini parmi les quatre,
     alors rechercher le sommet le moins précis: soit (k,j) son indice;
          calculer la valeur val de ce sommet en fonction des trois autres;
          trouvé:=précision de val meilleure que celle de ^l θ^j_k ;

            si trouvé alors amélioration:=vrai; retour fin de si;
       fin de si;
    fin de pour;
fin de pour;

--2°recherche d'une inversion non saturée:

pour l1=1 à p faire
     pour (k,j)∈ Π^{l1,1}_n faire
       si (l1+k impair) alors
        si (au moins l'une des valeurs ^{l1}θ^j_k ou ^{l1-1}θ^j_k est définie)

         alors rechercher la valeur la moins précise des deux:
                   soit l son indice, l' étant l'autre;
                calculer la valeur val en inversant ^{l'}θ^j_k;

                trouvé:= précision de val meilleure que celle de ^l θ^j_k;

                 si trouvé alors amélioration:=vrai; retour fin de si;
           fin de si;
        fin de si;
     fin de pour;
fin de pour;

--3° pas d'amélioration possible:

     amélioration:=faux;

fin d'amélioration;
```

3.2.5 Coût de cet algorithme.

Cet algorithme aveugle ne tient aucun compte des relations de proximité entre losanges ou couples de valeurs inverses l'une de l'autre. Chacun des losanges peut être testé plusieurs fois, chacune des valeurs peut être recalculée de nombreuses fois: non seulement cet algorithme risque d'être très coûteux, mais on n'est même pas certain qu'il converge, c'est à dire qu'il s'arrête au bout d'un temps fini. En réalité, on peut envisager une preuve de l'arrêt de cet algorithme reposant sur le fait que l'erreur sur le résultat ne peut être moindre que celle qui entache les données, mais nous n'avons pas développé cette question. Néanmoins, cet algorithme décrit assez bien la motivation suivante: exploiter chacune des identités qui interviennent dans ce problème pour améliorer la précision de la variable la plus grossière qui intervient dans cette identité. Bien entendu, l'idée que je présente ici dans le contexte particulier des algorithmes de losange peut s'adapter avec plus ou moins de succès à tous les algorithmes reposant sur un ensemble redondant de relations algébriques. L'exploitation conjuguée de cette redondance et des informations sur la précision avec laquelle chaque résultat intermédiaire est calculé fournira un algorithme d'une grande fiabilité numérique. Il ne faut toutefois pas perdre de vue le fait que le coût et par suite l'intérêt d'un tel algorithme dépendra beaucoup de la façon dont cette redondance aura été exploitée.

4 Programmation effective de l'algorithme.

4.1 Algorithme de principe effectif.

4.1.1 L'ordre principal.

L'algorithme de principe qui vient d'être présenté présente divers inconvénients:
1° son coût risque d'être très grand;
2° sa convergence n'est pas assurée;
3° il nécessite un très grand espace mémoire;
On réduira ces difficultés en s'imposant un ordre de balayage du tableau ${}^{l}\theta_{k}^{j}$ et en couplant la mise

en œuvre de la règle du losange avec celle des inversions. A cette fin, on se réfèrera à l'ordre principal suivant:

```
pour (chaque nouvelle valeur à traiter x_i) faire
--traitement d'une diagonale repérée par l'indice i:
   pour (chaque indice de colonne k) faire
    --traitement de la colonne d'indice k:
       pour (chaque indice de tableau l) faire
       --traitement de l'élément de tableau θ_k^(i-k) (l);
```

4.1.2 Exceptions.

4.1.2.1 Idée de base.

Cet ordre principal souffrira toutefois des exceptions qui seront clairement explicitées dans le texte du programme. L'idée directrice est la suivante: le traitement de l'élément ${}^{l}\theta_{k}^{i-k}$ commence

par le calcul de sa valeur au moyen de la relation du losange au sommet droit duquel il est situé; si l et k ont la même parité, alors l'élément ${}^{l}\theta_{k}^{i-k}$ est l'inverse de l'élément ${}^{l+1}\theta_{k}^{i-k}$ qui sera traité

ultérieurement: aucun test n'est encore possible; au contraire, si l et k sont de parité différente, ${}^{l}\theta_{k}^{i-k}$ est l'inverse de l'élément ${}^{l-1}\theta_{k}^{i-k}$ qui a déjà été calculé et on va donc pouvoir utiliser la rela-

tion d'inversion qui les lie: si le nouvel élément ${}^{l}\theta_k^{i-k}$ est plus précis que le précédent, on recalculera une nouvelle valeur de ce dernier, puis, si cette nouvelle valeur est plus précise, on réutilisera la relation du losange qui avait servi à son élaboration pour recalculer une des valeurs à partir de laquelle il avait été initialement calculé, le processus pouvant être itéré ... Par contre, si le nouvel élément ${}^{l}\theta_k^{i-k}$ est moins précis que le précédent, on recalculera une nouvelle valeur de ${}^{l}\theta_k^{i-k}$ par inversion de ${}^{l-1}\theta_k^{i-k}$, puis si cette nouvelle de ${}^{l}\theta_k^{i-k}$ est plus précise que celle qui avait été calculée avec la relation du losange, on réutilisera cette relation du losange pour recalculer une des valeurs à partir de laquelle il avait été calculé, le processus pouvant être itéré.

4.1.2.2 Traitement de l'élément ${}^{l}\theta_k^{i-k}$.

Le traitement de l'élément ${}^{l}\theta_k^{i-k}$ comporte deux phases bien distinctes:

1° on commence par appliquer la règle du losange pour calculer le quatrième sommet ${}^{l}\theta_k^{i-k}$ du losange d'indice (k-1,i-k).
2° selon la parité relative de k et l, cette valeur est liée à l'élément homologue du tableau d'indice l+1 (si k et l ont même parité) ou du tableau d'indice l-1 (si k et l sont de parité différente). Deux cas se présentent alors:
2.1 les indices k et l ont même parité: l'élément ${}^{l+1}\theta_k^{i-k}$ n'ayant pas encore été calculé, nous attendrons que cet élément soit calculé par la relation du losange pour pouvoir comparer la précision des deux approximations. Le traitement secondaire s'arrête là.
2.2 les indices k et l ont une parité différente: l'élément ${}^{l-1}\theta_k^{i-k}$ ayant déjà été calculé, nous disposons de deux approximations calculées indépendamment l'une de l'autre qui sont théoriquement inverses l'une de l'autre. Trois cas peuvent se présenter, qui dépendent de la précision relative avec laquelle sont connus les deux éléments en question:
2.2.1 les deux valeurs sont connues avec la même précision: on ne peut exploiter aucune des deux valeurs pour améliorer l'autre. Là encore, arrêt du traitement secondaire.
2.2.2 la valeur ${}^{l}\theta_k^{i-k}$ est connue avec une meilleure précision que ne l'est la valeur ${}^{l-1}\theta_k^{i-k}$: on va profiter de cela pour tenter d'obtenir une meilleure approximation de ${}^{l-1}\theta_k^{i-k}$. La valeur $v = ({}^{l}\theta_k^{i-k})^{-1}$ est une nouvelle approximation de ${}^{l-1}\theta_k^{i-k}$. Si la précision de v n'est pas meilleure que celle de la valeur actuelle de ${}^{l-1}\theta_k^{i-k}$, on arrête le traitement secondaire, mais si la précision de v est meilleure que celle de ${}^{l-1}\theta_k^{i-k}$, alors la nouvelle valeur v peut et doit être substituée à l'ancienne. Cette ancienne valeur avait été calculée au moyen de la règle du losange à partir de trois valeurs; deux d'entre elles appartiennent à la diagonale antérieure, et leur valeur est figée, mais la troisième appartient à la diagonale actuelle, et sa valeur peut éventuellement être corrigée. On recalculera donc une nouvelle approximation w de ${}^{l-1}\theta_{k-1}^{i-k+1}$ en réutilisant la relation du losange centré en et on comparera la précision de cette nouvelle valeur à celle de l'ancienne.

4.2 Technique de la diagonale montante.

4.2.1 Quelques notations.

L'algorithme sera programmé selon la technique des diagonales montantes, ce qui permet de traiter successivement les valeurs données, mais cette particularité n'est pas exploitée dans la présentation du programme puisque nous irons chercher les valeurs dans un fichier où elles auront été préparées. Lors de l'introduction de la donnée numéro i, i.e. le triplet (x(i),t(i-1),s(i-2)), on dispose déjà d'un tableau u(j,k) contenant les éléments u(j,l) de la dernière diagonale calculée: $u(j,l)=\theta_{i-1-l}^{j}$, où les indices j et l repèrent respectivement le numéro du tableau (de 0 à jmax) et le premier indice des données utilisé pour calculer la valeur concernée. A cette diagonale numéro i-1 vient s'adjoindre une nouvelle diagonale numéro i dont on rangera les valeurs dans le tableau v(j,k) en respectant la correspondance: $v(j,l)=\theta_{i-1}^{j}$. Bien sur, il serait relativement aisé d'éviter l'introduction d'une seconde diagonale, mais cela compliquerait encore un peu un algorithme dont la principale qualité n'est certes pas la simplicité.

4.2.2 Algorithme général.

1) initialisation: $u(0,0)=x_0$;

2) boucle générale:

pour chaque nouvel élément (i=1 à n) faire

2.1) actualisation des coefficients (§4.2.3);
2.2) initialisations propres à la diagonale montante n°i (§4.2.4);
2.3) traitement principal:

pour chaque colonne (k=1 à i) faire
2.3.1) pour chaque tableau (j=0 à min(k-1,jmax)) faire

2.3.1.1) appliquer la règle du losange:
$v(j,i-k)=u(j,i-k+1)+d(i-k)/(v(j,i-k+1)-u(j,i-k))$;
2.3.1.2) selon la parité, comparer ce résultat à l'élément homologue du tableau dont l'indice est immédiatement inférieur, et, s'il y a lieu, améliorer la précision de certains résultats antérieurs (§4.2.5);
fin pour j;

2.3.2) s'il y a lieu, initialisation du tableau n°k (§4.2.6);
fin pour k;

2.4) actualisation du tableau u pour le traitement de la diagonale suivante, et copie de la diagonale actuelle dans le tableau final (§4.2.7).

fin pour i;

4.2.3 Actualisation des coefficients.

Cette actualisation des coefficients correspond à l'algorithme de losange général défini en [8], dans lequel on a: $a_k^{(i)}=\sigma_i+\sum_{j=0}^{k}(\tau_{i+j}-\sigma_{i+j})$. Les coefficients $d(j) = a_{1-j}^{j}$ de la nouvelle diagonale sont alors calculées en fonction des coefficients $c(j)=a_{1-j-1}^{j}$ de l'ancienne de la façon suivante:

```
d(j-1)=t_{i-1};
si i>1 alors
      c(i-1)=s(i-2); écart=d(i-1)-c(i-1);
    pour j=2 à n faire  d(i-j)=c(i-j)+écart;
fsi;
```

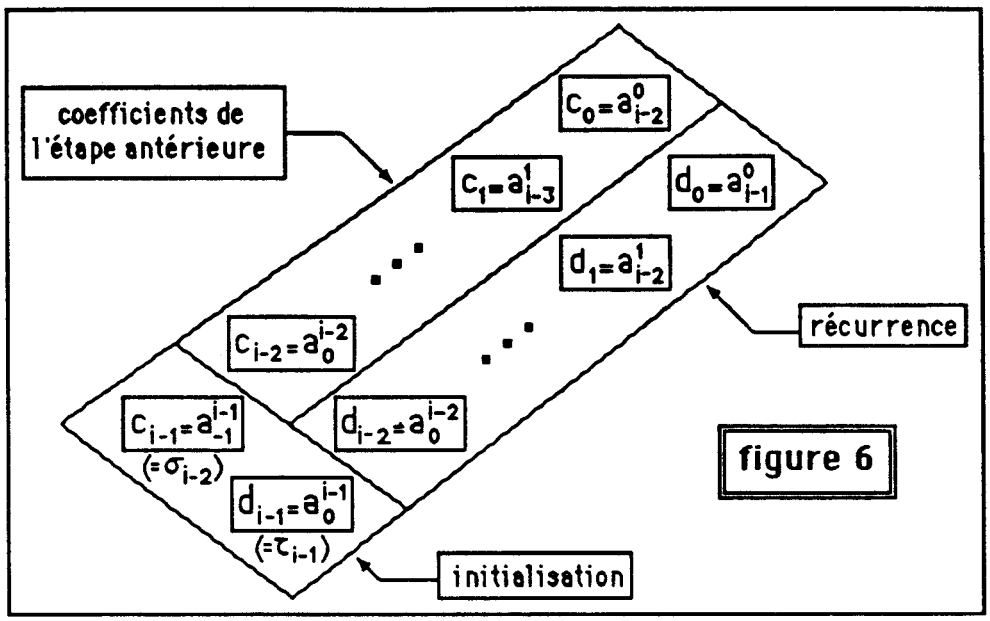
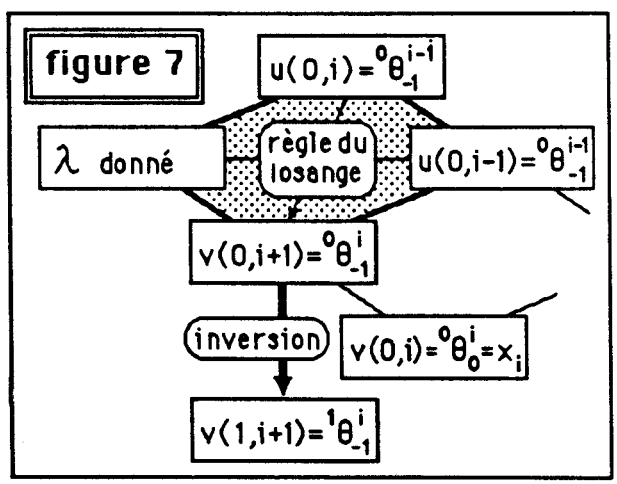


figure 6

En réalité, l'opportunité que nous venons de décrire n'a pas été exploitée dans le programme fortran qui suit. La version actuelle du programme fortran que nous annexons au présent travail fait appel à une variante beaucoup moins souple n'autorisant que la mise en œuvre des deux algorithmes de losange les plus courants (ε-algorithme et ρ-algorithme).

4.2.4 Initialisation de la diagonale n°i.



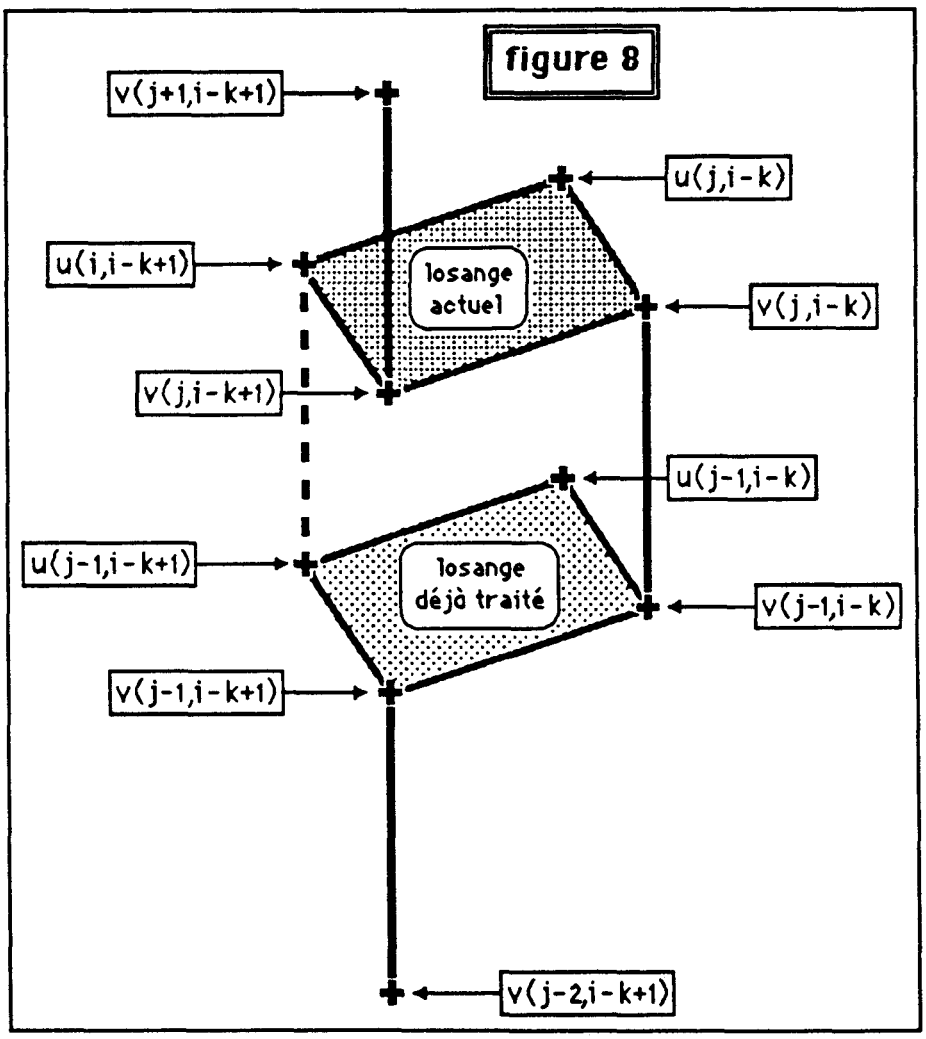
On calcule les premiers éléments des tableaux n° 0 et 1 au moyen de:

```
  v(0,i+1)=u(0,i)+c(i-1)/(u(0,i-1)-λ); (règle du losange)
v(0,i)=x_i;  u(0,i)=v(0,i+1);
  v(1,i+1)=1/v(0,i+1).
```

4.2.5 Comparaison de la précision.

Il s'agit là d'une opération délicate. La consultation des figures 8 ,9 et 10 permet d'en suivre plus commodément l'articulation.

La figure 8 résume la situation initiale dans le cas où j et k ont la même parité: elle rassemble 10 éléments qui sont liés par 6 relations. Quatre d'entre eux, u(j,i-k), u(j,i-k+1), u(j-1,i-k et u(j-1,i-k+1) ont été calculés lors de l'étape antérieure: leur valeur est donc indépendante de la dernière donnée introduite, à savoir x_i, et le traitement actuel, qui repose précisément sur la connaissance de x_i ne peut pas l'améliorer. Il n'en va pas de même pour les six autres, u(j-2,i-k+1), u(j-1,i-k), u(j-1,i-k+1), v(j,i-k), v(j,i-k+1) et v(j+1,i-k+1). Notons tout d'abord que parmi ces valeurs, du moins parmi celles qui existent car certaines d'entre elles peuvent ne pas être définies, il n'en est qu'une qui n'a point encore été calculée: u(j,i-k).



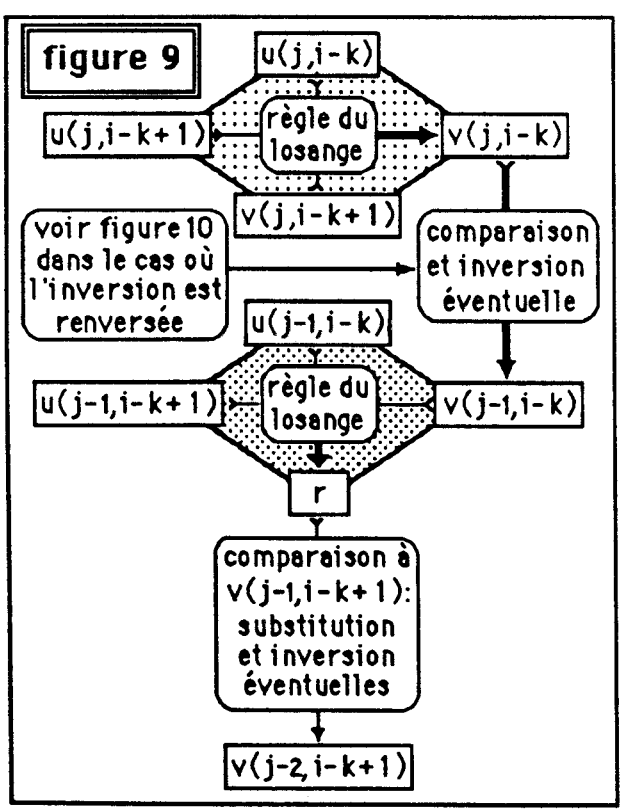
Ces dix valeurs sont liées par six relations que nous rappelons brièvement:
1° La règle du losange intervient deux fois: elle lie celles du tableau numéro j, d'une part, et celles du tableau numéro j-1 d'autre part;
2° La règle d'inversion intervient quatre fois, puisqu'elle lie les composants de chacun des couples suivants:

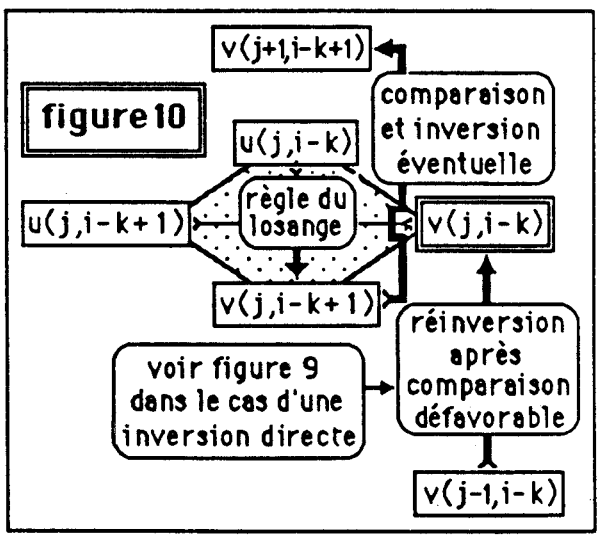
(v(j-2,i-k+1),v(j-1,i-k+1)),
(v(j-1,i-k),v(j,i-k)),
(v(j,i-k+1),v(j+1,i-k+1)),
(u(j-1,i-k+1),u(j,i-k+1));

Les quatre relations qui ne font pas intervenir v(j,i-k) ont certes déjà été exploitées, mais cela ne signifie pas qu'elles ne le seront plus: si une relation a déjà été exploitée, tous ses éléments ont déjà été calculés; si nous disposons d'une autre relation pour calculer l'un de ces éléments, nous connaissons alors deux approximations de cet élément. Si la nouvelle approximation est plus précise que l'ancienne, nous retiendrons cette nouvelle valeur, mais notre traitement ne va pas s'arrêter là: la plus mauvaise des deux approximations avait été calculée au moyen d'une relation que nous pouvons utiliser pour recalculer le moins précis de ses éléments en fonction des autres.

Au contraire, si l'ancienne relation est plus précise que la nouvelle, nous conserverons bien entendu cette ancienne valeur, mais, là encore, notre traitement n'est pas achevé pour autant: si le résultat de la relation que nous venons d'exploiter n'est pas précis, c'est qu'au moins une des données était plus imprécise que le résultat que nous avions obtenu en utilisant une autre relation; s'il n'y a qu'une telle donnée imprécise, il est alors possible de réutiliser notre relation actuelle pour recalculer cette donnée imprécise à partir du résultat plus précis obtenu par un autre moyen.

Cette technique sera utilisée ici avec la restriction suivante: nous nous interdisons de recalculer les élément de la diagonale antérieure, c'est à dire les u(j,i-k); cela a deux conséquences immédiates. D'une part, la relation d'inversion qui lie u(j,i-k+1) et u(j-1,i-k+1) ne sera pas exploitée (elle l'a été lors de l'étape antérieure); d'autre part, dans la règle d'inversion comme dans celle du losange, il n'y a que deux valeurs du tableau u: chacune de ces valeurs peut alors être calculée en fonction de l'autre au moyen de cette relation et cela a pour effet de définir simplement l'élément d'une relation qu'on va pouvoir recalculer dès qu'on voit que l'utisation directe de cette relation conduit à un résultat moins précis que celui qu'on a obtenu par une autre relation.



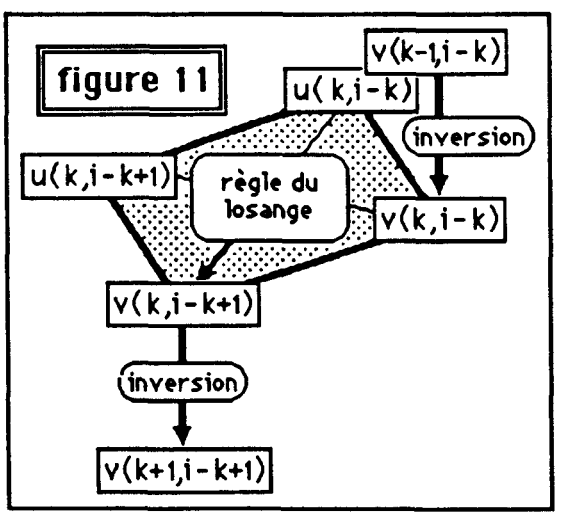


```
si (j≥1 et j-k pair) alors   --le dernier élément calculé v(j,i-k)= ^j θ_k^{i-k} est lié à l'élément
                                --homologue du tableau d'indice j-1; comparons les:
    si (v(j,i-k) plus précis que v(j-1,i-k)) alors  -- voir figure 9
      v(j-1,i-k)=1/v(j,i-k); --correction de la valeur la moins précise
         r=u(j-1,i-k+1)+c((v(j-1,i-k)-u(j-1,i-k+1));  -- utilisation verticale de
     -- la règle du losange pour recalculer r= ^{j-1} θ_{k-1}^{i-k+1} à partir duquel v(j-1,i-k)= ^{j-1} θ_k^{i-k}

      -- a déjà été calculé; comparons la précision des deux valeurs concurrentes:
         si (r plus précis que v(j-1,i-k+1) alors
          v(j-1,i-k+1)=r; -- correction de la valeur antérieure:
         -- comparer à l'élément homologue du tableau d'indice inférieur (s'il existe):
              si (j≥2 et alors si v(j-1,i-k+1) plus précis que v(j-2,i-k+1)) alors
              v(j-2,i-k+1)=1/v(j-1,i-k+1);-- correction de la valeur antérieure:
           fsi;
       fsi;
    sinon si (v(j,i-k) moins précis que v(j-1,i-k)) alors  --voir figure 10
      v(j,i-k)=1/v(j-1,i-k); -- correction de la valeur la moins précise:
         r=u(j,i-k+1)+c((v(j,i-k)-u(j,i-k+1));  -- utilisation horizontale de la
      -- règle du losange pour recalculer r= ^j θ_k^{i-k+1} à partir duquel v(j,i-k)= ^j θ_k^{i-k} avait

      -- été antérieurement calculé: comparons la précision des deux valeurs concurrentes:
         si (r plus précis que v(j,i-k+1) alors
          v(j,i-k+1)=r; -- correction de la valeur antérieure:
         -- comparer à l'élément homologue du tableau d'indice supérieur (s'il existe):
              si (j<i et alors si v(j,i-k+1) plus précis que v(j+1,i-k+1))  alors
              v(j+1,i-k+1)=1/v(j,i-k+1);-- correction de la valeur antérieure:
           fsi;
       fsi;
   fsi;
fsi;
```

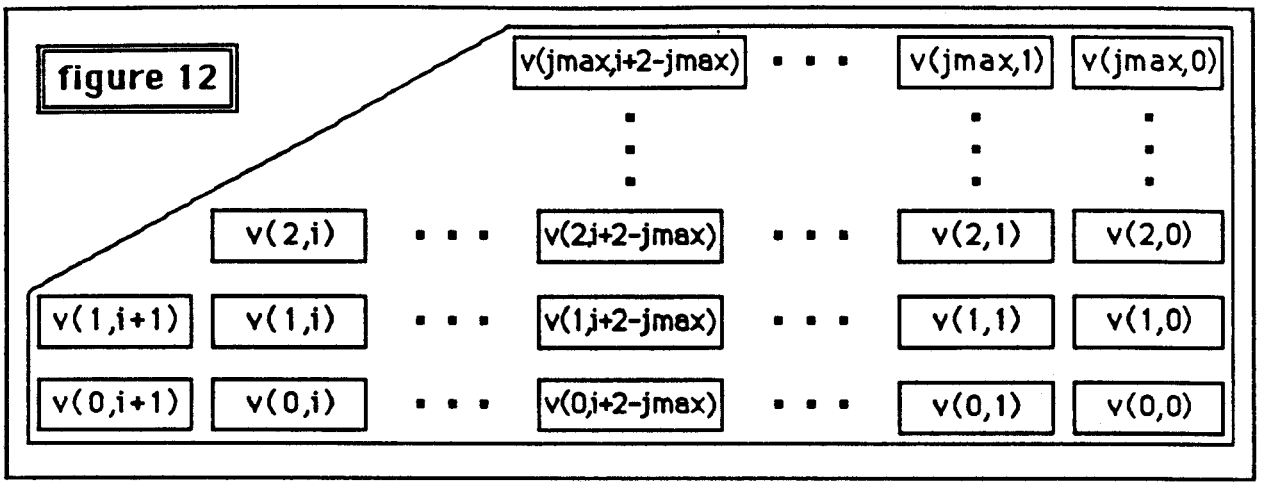
4.2.6 Initialisation du tableau numéro k.




```
si (k≤jmax) alors
      v(k,i-k)=1/v(k-1,i-k);
         v(k,i-k+1)=u(k,i-k)+d(i-k)/(v(k,i-k)-u(k,i-k+1));
    si (k<jmax) alors
          v(k-1,i-k+1)=1:v(k,i-k+1);
   fsi;
fsi;
```

4.2.7 Actualisation du tableau u et copie dans le tableau final.

Le tableau v a la structure suivante:



Après avoir recopié le tableau v dans le tableau u, nous initialisons les colonnes ayant l'autre parité dans le prochain tableau créé, s'il existe. Pour l'affichage, on se contente de recopier les valeurs du niveau zéro, en se limitant aux indices non négatifs:

```
pour chaque colonne (pour l:=0 à i+1) faire
    pour chaque niveau (pour j:=0 à min(i-2-l,jmax)) faire
          u(j,l):=v(j,l);
    fpour j;
fpour l;
si (i<jmax) alors u(i+1,0):=0  fsi ;
pour l:=0 à i faire
      théta(i-1,l):=v(0,l)
fpour l;
```

4.3 Notes sur le codage fortran.

Pour traduire cet algorithme en un programme fortran, j'ai adopté les règles suivantes:
Les nombres élémentaires sont des p-uplets (ou multinombres) rangés dans des tableaux. Rien ne nous interdit d'utiliser des variables indicées dont les éléments sont déjà de tels p-uplets. Toutefois, compte tenu du mode de rangement des tableaux fortran, l'indice repèrant les diverses composantes d'un tel p-uplet sera nécessairement le premier si nous souhaitons que les diverses composantes d'un multinombre soient contiguës (ce qui est essentiel si l'on veut que les opérations arithmétiques agissent simplement sur l'ensemble des composants de chaque multinombre) Pour éviter les problèmes de repérage, nous avons choisi de renverser l'ordre initial des indices. De plus les deux tableaux u et v ont été rangées dans un même tableau diag: ainsi, l'élément diag(l,k,j,i) correspond-il à la composante numéro l du mutinombre représenté par l'élément u(j,k) si i=0 et v(j,k) si i=1.

La représentation par des multinombre est spécifiée dans les commentaires. Dans la version actuelle du programme, les nombres $\tau(i)$ et $\sigma(i)$ qui correspondent à l'algorithme de losange le plus général n'apparaissent pas: les coefficients a_k^i sont calculés directement à partir des données, mais cette procédure nous confine provisoirement à l'ε-algorithme ou au ρ-algorithme.

5. programme FORTRAN

```
************************************************************************
*
*                          EPSILON et RHO-ALGORITHMES
*
************************************************************************
*
*       INVARIANCE HOMOGRAPHIQUE:
*
* Le programme utilise l'invariance homographique des colonnes de même parité en construisant systéma-
* tiquement plusieurs tableaux:
* Le premier est le tableau associé à la mise en œuvre classique de l'algorithme de losange retenu.
* Les éléments des colonnes d' indice impair du second sont les inverses des éléments homologues du pre-
* mier, ceux des colonnes d'indice pair du troisième sont les inverses des éléments de même indice du se-
* cond, etc...
* Le nombre de tableaux est défini par la valeur entière h, choisie par l'utilisateur dans l'intervalle (0-hh),
* où la constante entière hh est fixée par l'instruction parameter (hh=15 dans la version présentée).
* On profite du fait qu'à tout moment, un élément déterminé peut être calculé selon deux procédures (règle
* du losange ou inversion de l'élément homologue du tableau voisin) pour retenir celle qui fournit la valeur
* la plus fiable.
*
*          METHODE DE PERTURBATION:
*
* Cette fiabilite est appréciée au moyen de la méthode de perturbation de Vignes et Laporte qui est mise en
* œuvre ici dans les conditions suivantes:
* à chaque variable sont associées p valeurs distinctes dont la fonction respective est:
* la première valeur correspond au calcul normal de la valeur de la variable;
* chacune des p-2 suivantes correspond à une mise en œuvre indépendante de la méthode de perturbation:
* pour chaque opération,la i-ème composante sera calculée au moyen de la i-ème composante de chacun
* des opérandes, le dernier bit du résultat étant ensuite modifié ou non selon la valeur d'une procédure
* aléatoire à résultat booléen;
* la dernière valeur représente une estimation de l'erreur dont est entachée la composante principale (la
* première) à partir des valeurs des composantes perturbées (n° 2 à p-1). Le mode de calcul sera décrit
* plus loin.
* Le nombre p des composantes associées à la représentation d'un nombre (y compris l'estimation de
* l'erreur) est une valeur entière choisie par l' utilisateur dans l'intervalle (3..pp), où la constante entière
* pp est fixée par l'intruction parameter (pp=10 dans la version présentée).
*
*       L'ARITHMETIQUE COMPACTIFIEE UTILISEE:
*
* L' arithmétique flottante accessible à partir du type real*8 du fortran a été modifiée par l'introduction
* de deux valeurs externes: l'infini et l'indéfini. Pour que l'infini soit l'inverse de zéro, il a fallu introduire
* une valeur fictive de ce nom. Le champ des valeurs positives utilisable a été restreint à l'intervalle
* 16↑(-14)..16↑14.Toute valeur de module inférieur est réputée nulle, toute valeur de module supérieur
* est réputée infinie (à l'exception de l'indéfini qui sera codé avec une telle valeur). Les seuils 16↑(-14) et
* 16↑14 sont représentés par les variables mini et maxi de type real*8. Les valeurs spéciales sont codées
* par: zéro:16↑(-28),  infini:16↑28 et  indéfini:-16↑28. Leur valeur (non protégée par parameter) est
* accessible au moyen d'un commun nommé CONST.
*
* La droite numérique compactifiée par l'adjonction du point à l'infini est munie de la distance riemannienne
* qui permet de banaliser le rôle de ce point a l'infini. C'est cette distance qui nous permettra d'apprécier
* la qualité d'une approximation.
*
*
```

Annexe6: une version fiable et stable des algorithmes de losange.

```
*       PROGRAMME PRINCIPAL
*
* Ce programme va chercher les données dans le fichier "données" au moyen du sous-programme
* DONNEES, et met en œuvre l'algorithme de losange modifié atta-ché à ces données par l'appel du sous-
* programme LOSANGE. Le résultat est un tableau lostab que l'on corrige au moyen d'une utilisation
* inverse de l'algorithme du losange simple mise en oeuvre dans le sous-programme CORLOS. Les données
* initiales sont alors transformées (sous-programme INVDON) au moyen d'une inversion dont le pôle est
* choisi de facon à limiter les erreurs numériques (sous-programme CHXPOLE) et l'algorithme de losange
* modifié (LOSANGE) est appliqué à ces nouvelles données. Le tableau obtenu los1 est alors corrigé par
* appel à CORLOS, ses colonnes d'indice impair sont réinversées (appel au sous-programme INVLOS) pour
* donner un tableau los2 dont on complètera les colonnes d'indice impair par appel au sous-programme
* COMLOS (cette version provisoire n'est guère fiable: on lui préfèrera une mise en oeuvre du sous-
* programme LOSANGE à partir d'une initialisation des colonnes d'indice -1 et 1, comme cela sera fait dans
* les versions ultérieures).
* Nous disposons alors de deux tableaux lostab et los2 dont les valeurs (théoriquement égales) sont
* accompagnées d'une estimation de l'erreur qui entache chacun de leurs éléments. Nous pouvons donc
* effectuer trois contrôles simples et indépendants:
* *précision du premier calcul par consultation de la dernière composante de chaque valeur de lostab
* *précision du second calcul par consultation de la dernière composante de chaque valeur de los2
* *cohérence des deux calculs en calculant la distance des éléments correspondants des deux tableaux.
* Pour rendre ces résultats plus parlants, nous les avons directement traduits en nombre de chiffres
* décimaux attachés à un résultat (corrects par appel à NBCHLOS dans les deux premiers cas, communs
* aux deux représentations par appel à DISTLOS dans le dernier).
*
*       DECLARATIONS:
*
      implicit real*8 (a-z)
      common /const/ zero,infini,indefini,mini,maxi
      common /hasard/ b,nbt,nbv
      common /loterie/ num
      integer n,nn,i,b,nbt,nbv,p,pp,hh,h,num,large
      character*12 entree,sortie,brouillon,accord
      logical epsilon
      parameter (nn=50,pp=5,hh=20)
      dimension x(pp,0:nn),y(pp,0:nn),tt(pp,0:nn)
      dimension diag(pp,0:nn,0:hh,0:1),theta(-1:nn)
      dimension lostab(pp,0:nn,0:nn),los1(pp,0:nn,0:nn)
      dimension los2(pp,0:nn,0:nn),los3(pp,0:nn,0:nn)
      dimension lambda(pp),mu(pp),pole1(pp),man(pp),ligne(0:nn)
*
*       FONCTION DES VARIABLES NON INDICEES:
*
* *les cinq variables de type real*8 du commun CONST caractérisent l'arithmétique utilisée;
* *les trois variables entières du commun HASARD ne servent que dans le sous-programme ALEA;
* *les trois variables entières nn, pp et hh, initialisées et protégées par l'instruction parameter
*   fournissent les bornes maximales des trois dimensions laissées au choix de l'utilisateur:
*     nn: taille maximale du problème (les données sont indicables de 0 à nn).
*     hh: indice maximal des tableaux construits (indicables de 0 à hh).
*     pp: taille maximale des nombres manipulés (comportant:le nombre,ses perturbations,son erreur).
* *les trois variables n, p et h, associées aux précédentes, définissent la dimension réelle du problème:
*     n, nombre de données effectives du problème, est lu avec les données;
*     h et p, paramètres de la méthode de résolution, sont laissés au choix de l'utilisateur; attention, le
*      coût de l'algorithme est sensiblement proportionnel au produit (h+1)*(p-1)*(n+1)↑2.
* *le booléen epsilon lu dans le fichier des données, prend la valeur vraie si l'algorithme de losange est
*     l'epsilon-algorithme et faux dans le cas où c'est le rho-algorithme (cette version ne permet pas de
*     traiter les autres algorithmes de losanges);
*
```

```
*        FONCTION DES VARIABLES INDICEES:
*
**Les variables indicées de ce programme principal sont naturellement indicées par nn (taille du problème)
*  et pp (taille des nombres utilisés); tout nombre est représenté au moyen d'un tableau indicé par 1..pp,
*  mais seules les p premières composantes sont effectivement utilisées: tel est le cas des variables
*  lambda,mu et pole; lambda et mu représentent la valeur affectée à la colonne d'indice -2 dans le tableau
*  en losange, pole est la valeur retenue pour appliquer une nouvelle fois l'algorithme de losange.
**d' autres tableaux ont des fonctions plus spécifiques:
*  1°lostab,los1,los2 et los3 servent à mémoriser le résultat global de la mise en œuvre de l'algorithme de
*  losange; on a tenu compte de l'organisation particulière des tableaux en fortran pour placer en queue les
*  indices correspondant à une fonction plus globale; on trouve donc dans l'ordre: le repère de composante,
*  l'indice de diagonale descendante, puis l'indice de colonne; les colonnes sont ainsi rangées les unes der-
*  rière les autres;
*  2°les tableaux x,y,z et éventuellement tt contiennent les suites de valeurs auxquelles est appliqué le
*  sous-programme LOSANGE; tt contient la suite des abscisses dans le cas du rho-algorithme; le premier
*  indice repère les composantes de chaque nombre tandis que le second identifie le terme de la série;
**les trois derniers tableaux n'apparaissent dans le programme principal que pour éviter le surdimension-
*  nement des tableaux dans les sous-programmes dont ils devraient être des variables locales: ligne ne
*  sert que dans NBCHLOS, theta dans CREPOLE et diag dans LOSANGE.
*
*        CHOIX DES FICHIERS D'ENTREE-SORTIE ET DES PARAMETRES:
20    write (*,*) "nom du fichier d'entree (12 lettres au plus) "
      read (*,*) entree; open (3,file=entree,status="old")
      write (*,*) "nom du fichier de sortie principal (12 lettres)"
      read (*,*) sortie; open (2,file=sortie)
      write (*,*) "nom du fichier de sortie secondaire (12 lettres) "
      read (*,*)  brouillon; open (1,file=brouillon)
      WRITE  (*,*) "les trois fichiers ouverts sont:"
      write (*,*) "   fichier de données:    ",entree
      write (*,*) "   fichier de resultats:  ",sortie
      write (*,*) "   fichier intermediaire: ",brouillon
      write (*,*) "Cela vous convient-il? Tapez OUI pour confirmer."
      read (*,*) accord; if (accord.ne."OUI") goto 20
21    write (*,*) "entrez la hauteur: entier de l'intervalle (0:20)"
      read (*,*) h
      if ((h.lt.0).or.(h.gt.hh)) then
        write (*,*) "La hauteur ne convient pas: proposez une autre hauteur!"
        goto 21
      end if
22    write (*,*) "entrez la taille de la simulation de perturbation: entier de l'intervalle (3:12)"
      read (*,*) p
      if ((p.lt.3).or.(p.gt.pp)) then
        write (*,*) "La taille de la simulation ne convient pas: proposez une autre taille!"
        goto 22
      end if
23    write (*,*) "entrez la largeur d'affichage: entier de l'intervalle (3:6) "
      read (*,*) large
      if ((large.lt.3).or.(large.gt.6)) then
        write (*,*) "la largeur d'affichage ne convient pas: proposez une autre largeur "
        goto 23
      end if
*  INITIALISATIONS NON PARAMETREES:
      b=123457954; nbt=0; nbv=0; num=1; call crecst
      WRITE (*,*) "zero=    ",zero,    "   infini=",infini
      WRITE (*,*) "indefini=",indefini,"   mini=  ",mini,"   maxi=",maxi
      call crevec (p,infini,lambda)
```

```
**********************
*  TRAITEMENT PRINCIPAL:    *
**********************
* 1) ACQUISITION DES DONNEES:
      call donnees (3,p,epsilon,n,tt,x);  call impdon (2,p,epsilon,n,tt,x)
* 2) ALGORITHME DE LOSANGE PRINCIPAL:
      call losange(p,lambda,diag,x,n,n+2,tt,lostab,epsilon,h)
      write (2,*); write (2,*) "tableau obtenu directement "
      call implos (p,n,lostab,.true.,large)
* 3) CORRECTION DU TABLEAU:
      call corlos (p,n,lostab,tt,epsilon)
      write (2,*) "tableau initial corrige"; write (2,*);
      call implos (p,n,lostab,.TRUE.,large)
* 4) ESTIMATION DE L'ERREUR:
      write (2,*) "precision du tableau initial corrige";
      call implos (p,n,lostab(p,0,0),.TRUE.,large) ; write (2,*)
* 5) ESTIMATION DU NOMBRE DE CHIFFRES EXACTS:
      write (2,*) "nombre de chiffres corrects du tableau corrige"
      write (2,*); call nbchlos (p,n,lostab(p,0,0),ligne,.TRUE.,2*large)
**********************
*  TRAITEMENTS DE CONTROLE: *
**********************
* 1) TRANSFORMATION DES DONNEES:
      call crepole (p,n,n+1,x,theta,pole); call crevec (p,pole,pole1)
      call invsqc (p,n,x,y,pole1); call invpole (p,lambda,mu,pole1)
* 2) ALGORITHME DE LOSANGE PRINCIPAL:
      call losange (p,mu,diag,y,n,n+2,tt,los1,epsilon,h)
      write (1,*) "tableau derive des donnees inversees"
      call implos (p,n,los1,.TRUE.,large)
* 3) CORRECTION DU TABLEAU INVERSE:
      call corlos (p,n,los1,tt,epsilon)
      write (2,*) "correction du tableau derive des donnees inversees"
      call implos (p,n,los1,.TRUE.,large)
* 4) REINVERTION DU TABLEAU INVERSE (COLONNES PAIRES)
      call invlos (p,n,los1,los2,pole1)
      write (2,*) "reinversion des colonnes paires du tableau corrige"
      call implos (p,n,los2,.FALSE.,large)
* 5) RECONSTITUTION DU TABLEAU (PRESQUE) COMPLET:
      call comlos (p,n,los2,tt,epsilon)
      write (1,*) "tableau reinverse complete"
      call implos (p,n,los2,.TRUE.,large)
* 6) ESTIMATION DE L'ERREUR SUR CE TABLEAU RECONSTITUE:
      write (2,*) "erreur sur le tableau reinverse" ; write (2,*)
      call implos (p,n,los2(p,0,0),.TRUE.,large)
* 7) NOMBRE DE CHIFFRES CORRECTS DU TABLEAU RECONSTITUE:
      write (2,*) "nombre de chiffres corrects du tableau reinverse"
      write (2,*); call nbchlos (p,n,los2(p,0,0),ligne,.TRUE.,2*large)
* 8) DIFFERENCE DES DEUX TABLEAUX (INITIAL CORRIGE ET REINVERSE COMPLETE)
      call diflos (p,n,los2,lostab,los3)
      write (1,*) "difference:"; call implos (p,n,los3,.TRUE.,large)
* 9) DISTANCE RIEMANNIENNE DES DEUX TABLEAUX:
      call distlos (p,n,los2,lostab,los3); write (2,*) "distance:"
      call implos (p,n,los3,.TRUE.,large) ; write (2,*)
* 10) NOMBRE DE CHIFFRES COMMUNS AUX DEUX TABLEAUX:
      write (2,*) "nombre de chiffres communs aux deux tableaux: "
      write (2,*); call nbchlos (p,n,los3,ligne,.TRUE.,2*large)
    stop
    end
```

```
**********************************************************************
*
*  CREATION OU INITIALISATION:
*
* DONNEES va lire les donnees du probleme  dans un fichier(parametre nfich) où il trouve le booléen
* epsalgo (type du problème),l'entier n (taille du probleme, et les donnees numeriques.(voir introduction).
* CRECST initilise les valeurs (non protegees) du commun constantes.
* CRENB cree une multivaleur vec à partir d'un real*8 nb
*
**********************************************************************
      subroutine donnees(nfich,p,epsalgo,n,tt,x)
      integer p,n,i,pp,nfich
      parameter (pp=5)
      real*8  x(p,0:n),tt(p,0:n),terme,coef
      logical epsalgo
      read (nfich,10) epsalgo; read (nfich,*) n
10    format(l6)
      if (epsalgo) then
         do 1 i=0,n
           read (nfich,*) terme
1          call crevec (p,terme,x(1,i))
      else
         do 2 i=0,n
           read (nfich,*) coef,terme
           call crevec (p,coef,tt(1,i)); call crevec (p,terme,x(1,i))
2        continue
      end if
      return
      end
**********************************************************************
      subroutine crecst
      implicit real*8 (a-z)
      common /const/ zero,infini,indefini,mini,maxi
      maxi=16.d0**14; mini=1.d0/maxi ;infini=maxi*maxi
      indefini=-infini ; zero=1.d0/infini
      return
      end
**********************************************************************
      subroutine crevec (p,nb,vec)
      integer p,i
      real*8 nb,vec(p),anb
      implicit real*8 (a-z)
      common /const/ zero,infini,indefini,mini,maxi
      integer i
      do 1 i=1,p-1
        anb=dabs(nb)
          if (anb.lt.mini) then
             nb=zero
          else if (anb.gt.maxi) then
             nb=infini
           end if
1       vec(i)=nb
      call ajuster(p,vec)
      return
      end
```

Annexe6: une version fiable et stable des algorithmes de losange.

```
*****************************************************************
*
*  ARITHMETIQUE DE BASE:
*
*  Cet ensemble de sous-programmes concerne les questions liées à l'arithmétique: méthode de perturba-
*  tion,arithmétique compactifiée et distance associée: DISTANCE calcule la distance d de deux real*8 a et
*  b, en tenant compte du fait que l'un ou l'autre peut être infini ou indéfini. ALEA fournit une valeur boo-
*  léenne aléatoire l en utilisant une valeur entière b transmise par le commun *     HASARD et initialisée
*  dans le programme d'appel. MODIFIER change le dernier bit de la mantisse du reel*8 r. PERTURBER fait
*  appel aux deux sous-programmes précédents pour modifier le dernier bit lorsqu'aléa lui donne le feu
*  vert. AJUSTER fait appel aux deux sous-programmes précédents pour harmoniser l'ensemble des per-
*  turbations: il garantit qu'au moins une des perturbations sera effective; en outre, il teste si l'une des
*  composantes du multinombre obtenu n'est pas trop voisine de zéro ou de l'infini, auquel cas cette compo-
*  sante sera forcée à cette valeur spéciale.
*
*  Les opérations arithmétiques qui suivent agissent sur les multinombres définis en préambule. Nous n'uti-
*  lisons l'arithmétique compacti fiée que pour représenter les composantes des multinombres, l'erreur
*  étant représentée par un real*8 (ce qui autorise une distance nulle). Les opérations de base sont deux
*  opérations binaires, la SOMME et le PRODUIT, et deux opérations unaires, l'INVERSE et l'OPPOSE, ce qui
*  permet de décrire la DIFFÉRENCE et le QUOTIENT. On leur a adjoint la COPIE (ou affectation).
*
*****************************************************************
      subroutine distance(a,b,d)
      implicit real*8 (a-z)
      common /const/ zero,infini,indefini,mini,maxi
      if ((a.eq.indefini).or.(b.eq.indefini)) then
         d=1.d0
      else if ((b.eq.infini).and.(a.ne.infini)) then
         d=1.d0/dsqrt(1.d0+a*a)
      else if ((a.eq.infini).and.(b.ne.infini)) then
         d=1.d0/dsqrt(1.d0+b*b)
      else if (((a.eq.infini).and.(b.eq.infini)).or.((a.eq.zero).and.(b.eq.zero)))  then
         d=2.d0*mini
      else
        d=dabs(a-b)/dsqrt((1.d0+a*a)*(1.d0+b*b))
      end if
      return
      end
*****************************************************************
      subroutine alea(l)
      integer b,nbt,nbv
      common /hasard/ b,nbt,nbv
      logical l
      b=b*65539;  b=mod(b,2147483648)
      l=b.lt.0;  nbt=nbt+1
      if (l) nbv=nbv+1
      return
      end
*****************************************************************
      subroutine modifier (r)
      real*8 r,s
      integer*1 ent(8)
      equivalence (s,ent(1))
      s=r
      if (mod(ent(8),2).eq.0) then
         ent(8)=ent(8)+1
      else
         ent(8)=ent(8)-1
```

```
      endif
      r=s
      return
      end
*****************************************************************************
      subroutine perturber (r)
      real*8 r
      logical l
      call alea (l)
      if (l) call modifier (r)
      return
      end
*****************************************************************************
      subroutine ajuster (p,nb)
      implicit real*8 (a-z)
      dimension nb(1)
      integer i,num,p,q
      common /loterie/ num
      common /const/ zero,infini,indefini,mini,maxi
      q=p-1; num=mod(num,q)+1
      do 1 i=1,q
         if (i.eq.num) call modifier (nb(i))
          if (i.ne.1) call perturber (nb(i))
         anb=dabs(nb(i))
         if (anb.gt.maxi) nb(i)=infini
          if (anb.lt.mini) nb(i)=zero
1     continue
      nb(p)=mini
      do 2 i=2,q
         call distance (nb(1),nb(i),dist)
2        nb(p)=dmax1(nb(p),dist)
      call distance (nb(1),infini,dist)
      if (dist.le.2.d0*nb(p)) then
         nb(1)=infini
      else
         call distance (nb(1),zero,dist)
         if (dist.lt.2.d0*nb(p))  nb(1)=zero
      end if
      return
      end
*****************************************************************************
      subroutine somme(p,a,b,s)
      implicit real*8 (a-z)
      integer p,i
      common /const/ zero,infini,indefini,mini,maxi
      dimension a(1),b(1),s(1)
      external distance,perturber
      if ((a(p).eq.1.d0).or.(b(p).eq.1.d0)) then
         s(1)=indefini ; s(p)=1.d0 ; return
      end if
      do 1 i=1,p-1
         if ((a(i).eq.infini).and.(b(i).eq.infini)) then
            s(1)=indefini ; s(p)=1.d0 ; return
         else if ((a(i).eq.infini).or.(b(i).eq.infini)) then
             s(i)=infini
          else
             s(i)=a(i)+b(i)
          end if
```

```
1    continue
     call ajuster (p,s)
     return
     end
*************************************************************************
     subroutine produit (p,a,b,s)
     implicit real*8 (a-z)
     integer p,i
     common /const/ zero,infini,indefini,mini,maxi
     dimension a(1),b(1),s(1)
     external distance,perturber
     if ((a(p).eq.1.d0).or.(b(p).eq.1.d0)) then
        s(1)=indefini ; s(p)=1.d0 ; return
     end if
     do 1 i=1,p-1
       if (((a(i).eq.infini).and.(b(i).eq.zero)).or.((a(i).eq.zero).and.(b(i).eq.infini)))   then
          s(1)=indefini ; s(p)=1.d0 ; return
       else if ((a(i).eq.infini).or.(b(i).eq.infini)) then
           s(i)=infini
        else
           s(i)=a(i)*b(i)
        end if
1    continue
     call ajuster (p,s)
     return
     end
*************************************************************************
     subroutine copie (p,a,r)
     integer p,i
     real*8 a(1),r(1)
     do 1 i=1,p
1       r(i)=a(i)
     return
     end
*************************************************************************
     subroutine inverse(p,a,r)
     implicit real*8 (a-z)
     integer i,p
     common /const/ zero,infini,indefini,mini,maxi
     dimension a(1),r(1)
     if (a(p).eq.1.d0) then
        r(1)=indefini ; r(p)=1.d0
     else
        do 1 i=1,p-1
          aa=dabs(a(i))
          if (aa.gt.maxi) then
              r(i)=zero
          else if (aa.lt.mini) then
              r(i)=infini
           else
              r(i)=1.d0/a(i)
           end if
1       continue
        call ajuster (p,r)
     end if
     return
     end
*************************************************************************
```

```
      subroutine oppose(p,a,r)
      implicit real*8 (a-z)
      integer i,p
      common /const/ zero,infini,indefini,mini,maxi
      dimension a(1),r(1)
      if (a(p).eq.1.d0) then
         r(1)=indefini ; r(p)=1.d0
      else
         do 1 i=1,p-1
           aa=dabs(a(i))
           if (aa.gt.maxi) then
               r(i)=infini
           else if (aa.lt.mini)then
               r(i)=zero
            else
               r(i)=-a(i)
            end if
1        continue
         r(p)=a(p)
      end if
      return
      end
*****************************************************************************
      subroutine difference(p,a,b,r)
      implicit real*8 (a-z)
      integer pp,p
      parameter (pp=5)
      dimension a(1),b(1),r(1),c(pp)
      call oppose(p,b,c); call somme(p,a,c,r)
      return
      end
*****************************************************************************
      subroutine quotient(p,a,b,r)
      implicit real*8 (a-z)
      integer pp,p
      parameter (pp=5)
      dimension a(1),b(1),r(1),c(pp)
      call inverse(p,b,c); call produit(p,a,c,r)
      return
      end

*****************************************************************************
*
*    TRAITEMENT PRINCIPAL:
*
* Il est assuré par le sous-programme LOSANGE qui fait lui-même appel aux sous-programmes LOSHOR et
* LOSVER; ces deux modules correspondent à l'utilisation de la règle du losange: LOSHOR calcule la valeur
* de l'élément le plus à droite du losange en fonction des valeurs aux trois autres sommets et de la valeur
* du coefficient c, le résultat étant rangé correctement dans le tableau diag;  LOSVER calcule la valeur de
* l'élément situé en bas du losange en fonction de la valeur des trois autres sommets et du coefficient c, le
* résultat étant rangé dans un paramètre auxiliaire (res). La cohérence est assurée par les paramètres p,
* n et n2 pour les dimensions, et par i et j pour la localisation du losange dans le tableau diag.
*
* Le sous-programme LOSANGE est le cœur de ce programme: nous nous contenterons de commentaires
* succincts et nous renvoyons le lecteur au paragraphe 4  pour de plus amples explications.
*
*****************************************************************************
```

Annexe6: une version fiable et stable des algorithmes de losange.

```
      subroutine  losver(p,diag,i,j,c,n,n2,res)
      integer  n,j,k,i,pp,n2,hh,p
      parameter  (pp=5,hh=20)
      real*8  c(1),diag(p,0:n2,0:hh,0:1),d(pp),q(pp),res(1)
      call  difference  (p,diag(1,j-1,i,1),diag(1,j,i,0),d)
      call  quotient  (p,c,d,q)
      call  somme  (p,diag(1,j-1,i,0),q,res)
      return
      end
*****************************************************************************
      subroutine  loshor(p,diag,i,j,c,n,n2,res)
      integer  n,i,j,k,p,n2,hh,pp
      parameter  (pp=5,hh=20)
      real*8  c(1),diag(p,0:n2,0:hh,0:1),d(pp),q(pp),res(1)
      call  difference  (p,diag(1,j+1,i,1),diag(1,j,i,0),d)
      call  quotient(p,c,d,q)
      call  somme  (p,diag(1,j+1,i,0),q,res)
      return
      end
*****************************************************************************
      subroutine  losange  (p,lambda,diag,x,n,n2,tt,lostab,epsilon,h)
      implicit  real*8  (a-z)
      integer  i,j,k,l,m,m1,n,deb,ii,n1,pp,p,n2,hh,h1,h
      parameter  (pp=5,hh=20)
      common  /const/  zero,infini,indefini,mini,maxi
      logical  epsilon
      dimension  x(p,0:n),tt(p,0:n),diag(p,0:n2,0:hh,0:1),lambda(1)
      dimension  lostab(p,0:n,0:n)
      real*8  un(pp),c(pp),man(pp),zero1(pp),res(pp),ref(pp)
*  INITIALISATION PRINCIPALE
*   CREATION DES MULTINOMBRES UN ET ZERO
         call  crevec  (p,1.d0,un)
         call  crevec(p,zero,zero1)
*   INITIALISATION DE LA PREMIERE DIAGONALE
         call  copie(p,lambda,diag(1,1,0,0))
         call  copie  (p,zero1,diag(1,0,0,0))
       do  1  i=0,n
*    TRAITEMENT DE LA DIAGONALE MONTANTE D'INDICE I
*      INITIALISATION DES COLONNES D'INDICE -2 ET 0 DANS LE TABLEAU D'INDICE 0
           call  copie  (p,x(1,i),diag(1,i,0,1))
           call  copie  (p,lambda,diag(1,i+2,0,1))
*      INITIALISATION DES COLONNES D'INDICE -2 ET 0 DANS LE TABLEAU D'INDICE 0
           if  (epsilon)  then
              call  copie  (p,un,c)
            else
              call  copie(p,zero1,c)
            end  if
*      REGLE DU LOSANGE VERTICAL ET INVERSION DU RESULTAT (POUR LE TABLEAU N°1)
           call  losver  (p,diag,0,i+1,c(1),n,n2,man)
           call  copie  (p,man,diag(1,i+1,0,1))
           call  inverse  (p,diag(1,i+1,0,1),diag(1,i+1,1,1))
            do  2  k=1,i
*         TRAITEMENT DE LA COLONNE N° K
              h1=min0(k-1,h)
               do  3  j=0,h1
*           TRAITEMENT  DU  TABLEAU  N° J
                 if  (epsilon)  then
                    call  copie  (p,un,c)
```

```
      else
        call difference (p,tt(1,i),tt(1,i-k),c)
      end if
*     REGLE DU LOSANGE HORIZONTAL
        call loshor (p,diag,j,i-k,c,n,n2,res)
        call copie (p,res,diag(1,i-k,j,1))
      if ((mod(j+k,2).eq.0).and.(j.ge.1)) then
*     INVERSION POSSIBLE DE L'ELEMENT CALCULE
        call inverse (p,res,man)
        call copie (p,diag(1,i-k,j-1,1),ref)
        if (man(p).lt.ref(p)/1.4d0) then
*       L'ELEMENT INVERSE EST MEILLEUR QUE L'ELEMENT ANTERIEUR:
*       IL LE REMPLACE, ET ON CONTINUE L'EXPLORATION;
          call copie(p,man,diag(1,i-k,j-1,1))
          if (epsilon) then
            call copie (p,un,c)
          else
            call difference (p,tt(1,i),tt(1,i-k),c)
          end if
*       MISE EN OEUVRE DU LOSANGE VERTICAL ET COMPARAISON DE PRECISION
          call losver (p,diag,j-1,i-k+1,c,n,n2,man)
          if (man(p).lt.diag(p,i-k+1,j-1,1)) then
*           REMPLACEMENT DE L'ANCIENNE VALEUR
            call copie(p,man,diag(1,i-k+1,j-1,1))
            if (j.ge.2) then
*             NOUVELLE INVERSION ET COMPARAISON
              call inverse (p,diag(1,i-k+1,j-1,1),man)
              if (man(p).lt.diag(p,i-k+1,j-2,1)) then
*               REMPLACEMENT DE L'ANCIENNE VALEUR
                call copie (p,man,diag(1,i-k+1,j-2,1))
                call loshor (p,diag,j-2,i-k,c,n,n2,res)
                if (res(p).lt.diag(p,i-k,j-2,1)) then
                  call copie (p,res,diag(1,i-k,j-2,1))
                end if
              end if
            end if
          endif
        else if (man(p).gt.ref(p)*1.4d0) then
*       L'ELEMENT ANTERIEUREMENT CALCULE EST PLUS PRECIS QUE LE NOUVEAU;
*       ON RECALCULE CE NOUVEL ELEMENT EN FONCTION DE L'ANCIEN ET ON EXPLOITE
*       CETTE NOUVELLE VALEUR POUR CORRIGER CELLE QUI ETAIT A L'ORIGINE DE
*       L'IMPRECISION EN APPLIQUANT VERTICALEMENT LA REGLE DU LOSANGE.
          call inverse (p,ref,man)
          call copie (p,man,diag(1,i-k,j,1))
          call losver (p,diag,j,i-k+1,c,n,n2,res)
          if (res(p).lt.diag(p,i-k+1,j,1)) then
*           LA VALEUR RECALCULEE EST PLUS PRECISE QUE CELLE QUI AVAIT DEJA ETE
*           CALCULEE: ELLE LA REMPLACE ET ON VA LA COMPARER A LA VALEUR
*           HOMOLOGUE DU TABLEAU VOISIN.
            call copie (p,res,diag(1,i-k+1,j,1))
            if (j.lt.h) then
*             LA PROFONDEUR DE LA SINGULARITE NE DEPASSE PAS LE SEUIL
              call inverse (p,res,man)
              if (man(p).lt.diag(p,i-k+1,j+1,1)) then
*               LA NOUVELLE VALEUR INVERSEE EST EFFECTIVEMENT PLUS PRECISE QUE
*               CELLE DONT NOUS DISPOSIONS JUSQU'A PRESENT: ELLE LA REMPLACE
                call copie (p,man,diag(1,i-k+1,j+1,1))
              end if
```

```
                    end if
                  end if
                end if
              end if
*           FIN DU TRAITEMENT DU J-IEME TABLEAU
3           continue
            if (k.le.h) then
*           INITIALISATION DU K-IEME TABLEAU
              call inverse (p,diag(1,i-k,k-1,1),diag(1,i-k,k,1))
              if (epsilon) then
                call copie (p,un,c)
              else
                call difference (p,tt(1,i),tt(1,i-k),c)
              end if
              call losver (p,diag,k,i-k+1,c(1),n,n2,man)
              call copie (p,man,diag(1,i-k+1,k,1))
              if (k.lt.h) then
                call inverse (p,diag(1,i-k+1,k,1),diag(1,i-k+1,k+1,1))
              end if
            end if
*         FIN DU TRAITEMENT DE LA K-EME COLONNE
2         continue
          if (i.lt.h) then
*         INITIALISATION ARBITRAIRE D'UN ELEMENT DU TABLEAU N°I (COLONNE DE PARITE
*         DIFFERENTE DE CELLE DES ELEMENTS INVERSES DU TABLEAU N° I-1).
            call copie (p,zero1,diag(1,0,i+1,1))
          end if
*         MISE A JOUR EN FIN DE TRAITEMENT D'UNE DIAGONALE MONTANTE
          h1=min0(h,i+1)
          do 5 k=0,h1
            do 5 j=0,i+2-k
*           ACTUALISATION DE LA NOUVELLE DIAGONALE MONTANTE
              call copie (p,diag(1,j,k,1),diag(1,j,k,0))
              if ((k.eq.0).and.(j.le.i)) then
            COPIE DU PREMIER NIVEAU DE LA DIAGONALE MONTANTE DANS LE TABLEAU
              COMPLET LOSTAB.
                call copie (p,diag(1,j,k,1),lostab(1,j,i-j))
              end if
5         continue
          call impdiag(p,n,n2,i,diag,h)
1     continue
      return
      end
```

Annexe6: une version fiable et stable des algorithmes de losange.

```
*****************************************************************
*
*  POST-TRAITEMENTS:
*
*  Ces sous-programmes ont pour objet les post-traitements qui permettent d'apprécier a posteriori la
*  qualité des résultats du programme de losange. Ils ne participent donc pas à la mise en œuvre de l'algo-
*  rithme mais seulement au contrôle de sa validité.  Dans ces sous-programmes, les paramètres entiers p
*  et n repèrent respectivement la taille des multinombres et celle de la suite x donnée.
*
* CREPOLE calcule la valeur pole la plus éloignée (au sens de la distance  riemannienne) de l'ensemble des
* valeurs de la suite x. Le tableau theta est mis en paramètre pour autoriser un dimensionnement semi-
* dynamique. INVSEQ transforme la suite x en une suite y au moyen d'une inversion dont le pole est le pa-
* rametre pole. INVPOLE calcule l'inverse r d'un multinombre a dans une inversion dont le pole est le para-
* metre pole. INVLOS transforme les colonnes d'indice pair du tableau lostab en un tableau losinv au moyen
* d'une inversion dont le pole est le parametre pole. DIFLOS et DISTLOS génèrent un tableau losdif dont les
* éléments sont respectivement la différence ou la distance des deux tableaux los1 et los2. CORLOS corri-
* ge le tableau los en appli quant la règle du losange à l'envers: pour chaque losange, il recalcule la valeur
* de l'élément situé en haut ou à gauche en fonction des valeurs *  des trois autres sommets; cela permet
* de corriger les lacunes liées à une mise en œuvre trop systématique de la technique de la diagonale mon-
* tante; contrairement à l'algorithme LOSANGE, cette correction ne porte que sur le tableau principal.
* COMPLOS complète le tableau dont les colonnes d'indice pair ont été obtenues indirectement (c'est à dire
*  en réinversant les colonnes d'indice pair du tableau construit à partir des termes de la suite préalable-
*  ment inversés). Cette méthode très peu fiable conduit à des résultats doûteux.
*
*****************************************************************
      subroutine crepole(p,n,n1,x,theta,pole)
      implicit real*8 (a-z)
      integer p,n,n1,i,j,j1
      common /const/ zero,infini,indefini,mini,maxi
      dimension theta(-1:n1),x(p,0:n)
      pi=4.d0*datan(1.d0); theta(-1)=-pi
      do 1 i=0,n
         if (x(1,i).eq.infini) then
            w=pi
         else
           w=2.d0*datan(x(1,i))
         end if
         j=i; j1=i-1
         while (w.lt.theta(j1))
           theta(j)=theta(j1); j=j1; j1=j-1
         repeat
         theta(j)=w
 1    continue
      max=0.d0;  theta(n+1)=pi
      do 2 i=0,n+1
         ecart=dabs(theta(i)-theta(i-1))
         if (ecart.gt.max) then
           max=ecart; imax=i
         end if
 2    continue
      maxtheta=(theta(imax)+theta(imax-1))/4.d0; pole=dtan(maxtheta)
      return
      end
*****************************************************************
```

```
      subroutine invpole (p,a,r,pole)
      integer pp,p
      parameter (pp=5)
      real*8  a(1),r(1),pole(1),dif(pp),inv(pp)
      call difference(p,a,pole,dif); call inverse (p,dif,inv)
      call somme (p,pole,inv,r)
      return
      end
*****************************************************************************
      subroutine invsqc (p,n,x,y,pole)
      integer p,n,i
      real*8  x(p,0:n),y(p,0:n),pole(1)
      do 1 i=0,n
         call invpole (p,x(1,i),y(1,i),pole)
1     continue
      return
      end
*****************************************************************************
      subroutine invlos(p,n,lostab,losinv,pole)
      real*8  lostab(p,0:n,0:n),losinv(p,0:n,0:n),pole(1)
      logical impair
      integer n,i,k,q,r,p
      do 2 i=0,2*n
         q=i/2;  r=i-2*q
         if (r.eq.0) then
            do 1 k=0,min0(q,n-q)
               call invpole (p,lostab(1,q-k,2*k),losinv(1,q-k,2*k),pole)
1           continue
         end if
2     continue
      return
      end
*****************************************************************************
      subroutine diflos(p,n,los1,los2,los)
      real*8  los1(p,0:n,0:n),los2(p,0:n,0:n),los(p,0:n,0:n)
      logical impair
      integer n,i,k,p
      do 3 i=0,n
         do 3 k=0,i
            call difference (p,los1(1,i-k,k),los2(1,i-k,k),los(1,i-k,k))
3     continue
      return
      end
*****************************************************************************
      subroutine distlos(p,n,los1,los2,los)
      real*8  los1(p,0:n,0:n),los2(p,0:n,0:n),los(p,0:n,0:n)
      integer n,i,k,p
      do 3 i=0,n
         do 3 k=0,i
            call distance (los1(1,i-k,k),los2(1,i-k,k),los(1,i-k,k))
3     continue
      return
      end
*****************************************************************************
```

```
      subroutine corlos(p,n,los,tt,epsilon)
      integer n,pp,p,i,k
      parameter (pp=5)
      logical epsilon
      real*8 los(p,0:n,0:n),tt(p,0:n),un(pp),dif(pp),quo(pp),res(pp),c(pp)
      call crevec(p,1.d0,un)
      do 1 i=n-1,2,-1
         do1 k=i,3,-1
*         CONTROLE DE LOS (k,i-k)
          if (k.lt.i) then
*          CONTROLE HORIZONTAL
            if (epsilon) then
               call copie (p,un,c)
             else
               call difference (p,tt(1,i+1),tt(1,i-k-1),c)
             end if
            call difference (p,los(1,i-k-1,k+1),los(1,i-k,k+1),dif)
            call quotient (p,c,dif,quo); call somme(p,los(1,i-k-1,k+2),quo,res)
             if (res(p).lt.los(p,i-k,k)) then
               call copie (p,res,los(1,i-k,k))
             end if
          end if
          if (k.gt.3) then
*          CONTROLE VERTICAL
            if (epsilon) then
               call copie (p,un,c)
             else
               call difference (p,tt(1,i+1),tt(1,i-k),c)
             end if
            call difference (p,los(1,i-k+1,k-1),los(1,i-k,k+1),dif)
            call quotient (p,c,dif,quo); call somme (p,los(1,i-k+1,k),quo,res)
             if (res(p).lt.los(p,i-k,k)) then
               call copie (p,res,los(1,i-k,k))
             end if
          end if
  1   continue
      return
      end

*****************************************************************************
      subroutine comlos(p,n,los,tt,epsilon)
      integer n,pp,i,k,p
      logical epsilon
      parameter (pp=5)
      real*8 los(p,0:n,0:n),tt(p,0:n), c(pp),dif(pp),quo(pp),res(pp),un(pp)
      call crevec(p,1.d0,un)
      do 1 i=1,n
         if (epsilon) then
            call copie (p,un,c)
          else
            call difference (p,tt(1,i),tt(1,i-1),c)
          end if
         call difference (p,los(1,i,0),los(1,i-1,0),dif)
         call quotient (p,c,dif,los(1,i-1,1))
         do 1 k=3,i,2
            if (epsilon) then
               call copie (p,un,c)
             else
```

```
            call difference (p,tt(1,i),tt(1,i-k),c)
          end if
          call difference (p,los(1,i-k+1,k-1),los(1,i-k,k-1),dif)
          call quotient (p,c,dif,quo)
          call somme (p,los(1,i-k+1,k-2),quo,los(1,i-k,k))
          if (k.lt.i) then
            if (epsilon) then
              call copie (p,un,c)
            else
              call difference (p,tt(1,i),tt(1,i-k-1),c)
            end if
            call difference (p,los(1,i-k-1,k+1),los(1,i-k,k-1),dif)
            call quotient (p,c,dif,quo); call somme (p,los(1,i-k-1,k),quo,res)
            if (res(p).lt.los(p,i-k,k)) then
              call copie (p,res,los(1,i-k,k))
            end if
          end if
 1    continue
      return
      end
*****************************************************************************
*
* SOUS-PROGRAMMES D'IMPRESSION:
*
* IMPDON imprime les données pour contrôle. IMPVEC imprime un multinombre r. IMPDIAG imprime toutes
* les informations relatives à la i-ème diagonale montante, pour les tableaux utilisés jusqu'au niveau h.
* IMPLOS imprime un tableau de losange complet (paramètre lostab).On se restreint aux colonnes paires en
* fixant le logical impair à .FALSE. NBCHLOS imprime aussi un tableau de losange complet; toutefois, il
* n'imprime pas les valeurs contenues * dans le tableau qui figure en parametre mais le nombre de chiffres
* décimaux exacts estimé de chacun des nombres de ce tableau (c'est en fait l'inverse du logarithme déci-
* mal de l'erreur qu'on trouve en p-ième composante de chaque multinombre).
*
*****************************************************************************
      subroutine impdon(nfich,p,epsalgo,n,tt,x)
      integer p,n,i,pp,nfich
      parameter (pp=5)
      real*8 x(p,0:n),tt(p,0:n),terme,coef
      logical epsalgo
      write (nfich,*) ; write (nfich,*) "données : taille du problème = ",n
      write (nfich,*)
      if (epsalgo) then
        write (nfich,*) " epsilon-algorithme "; write (nfich,*)
        write (nfich,101)
        do 1 i=0,n
          write (nfich,102) i,x(1,i),x(p,i)
 1      continue
      else
        write (nfich,*) " rho-algorithme "; write (nfich,*); write (nfich,103)
        do 2 i=0,n
          write (nfich,104) i,tt(1,i),x(1,i),tt(p,i),x(p,i)
 2      continue
      end if
      return
 101 format ("indice          terme            erreur")
 102  format (i5,d25.15,d15.5)
 103 format ("indice          abscisse          terme            err-coef   err-absc ")
 104  format (i5,2d25.15,2d15.5)
      end
```

Annexe6: une version fiable et stable des algorithmes de losange.

```
*****************************************************************************
      subroutine impvec (p,r)
      real*8  r(1)
      integer i,p
      write  (1,98)  r(1),(r(i)-r(1),i=2,p-1),r(p)
98    format  (d21.14,5d10.3,/,21x,5d10.3)
      return
      end
*****************************************************************************
      subroutine impdiag (p,n,n2,i,diag,h)
      integer  n,i,j,k,l,m,p,n2,hh,h1,h
      parameter (hh=20)
      real*8  diag(p,0:n2,0:hh,0:1)
      h1=min0(i-1,h)
      do 1 m=0,h1
         write (1,*) "tableau nO : ",i
         do 3 j=0,i-m
            write  (1,98)m,j,diag(1,j,m,0),(diag(l,j,m,0)-diag(1,j,m,0),
     1       l=2,p-1),          diag(p,j,m,0)
3        continue
1     continue
98      format (2i3,d21.14,5d10.3)
      return
      end
*****************************************************************************
      subroutine implos(p,n,lostab,impair,largeur)
      real*8  lostab(p,0:n,0:n)
      logical  impair
      integer  n,i,k,q,r,p,largeur,nbande,ibande,kdeb,kfin,kmax,kstop,kd
      write (*,*) "imp los"; write (2,100)
      nbande=n/largeur/2;  kfin=-1
      do 1 ibande=0,nbande
         kdeb=kfin+1;  kfin=kfin+largeur;  kd=min0(2*kfin+1,n)
         write  (2,101)  (k,k=2*kdeb,kd)
         do 2 i=0,2*n
            q=i/2;   r=i-2*q
            if (r.eq.0) then
               kmax=min0(kfin,q,n-q)
               write  (2,102)  i,(lostab(1,q-k,2*k),k=kdeb,kmax)
            else if (impair) then
               kstop=min0(kmax,n-q-1)
               write  (2,103)  (lostab(1,q-k,2*k+1),k=kdeb,kstop)
            end if
2        continue
         write  (2,100)
1     continue
      write  (2,100)
      return
100   format (/)
101   format (5x,25i8)
102   format (/i2,25d16.6)
103   format (/10x,25d16.6)
      end
*****************************************************************************
      subroutine nbchlos(p,n,lostab,ligne,impair,largeur)
      common /const/  zero,infini,indefini,mini,maxi
      real*8  lostab(p,0:n,0:n),ligne(0:n),val,zero,infini,indefini,mini
      real*8  maxi
```

Annexe6: une version fiable et stable des algorithmes de losange.

```
      logical  impair
      integer  n,i,k,q,r,p,largeur,nbande,ibande,kdeb,kfin,kmax,kstop,kd
      nbande=n/largeur/2;  kfin=-1
      do 1 ibande=0,nbande
         kdeb=kfin+1;  kfin=kfin+largeur;  kd=min0(2*kfin+1,n)
         write  (2,101)  (k,k=2*kdeb,kd);  write  (2,*)
         do  2  i=0,2*n
            q=i/2;   r=i-2*q
           if  (r.eq.0)  then
             kmax=min0(kfin,q,n-q)
             do  3  k=kdeb,kmax
                val=lostab(1,q-k,2*k)
                if  (val.lt.1.D-20)  val=1.D-20
               ligne(k)=-DLOG10(val)
3             continue
             write  (2,102)  i,(ligne(k),k=kdeb,kmax)
           else if  (impair)  then
             kstop=min0(kmax,n-q-1)
             do  4  k=kdeb,kstop
                val=lostab(1,q-k,2*k)
                if  (val.lt.1.D-20)  val=1.D-20
               ligne(k)=-DLOG10(val)
4             continue
             write  (2,103)  (ligne(k),k=kdeb,kstop)
           end  if
2        continue
         write  (2,100)
1     continue
      write  (2,100)
      return
  100 format (/)
  101 format (5x,25i5)
  102 format (i2,25f10.1)
  103 format (7x,25f10.1)
      end
```

6. Quelques résultats numériques

6.1 Présentation générale.

6.1.1 Motivation.

La présentation d'un tel travail ne prend son sens que si elle est illustrée par des exemples d'utilisation significatifs. Aussi avons nous retenu un certain nombre de cas dans lesquels l'ε-algorithme ou le ρ-algorithme sont mis en œuvre dans des conditions singulières ou quasi-singulières. Le souci de montrer que cette méthode n'était pas une curiosité sans intérêt réel nous a conduit à choisir des exemples de taille significative (jusqu'à plus de 18 valeurs interpolées pour le ρ-algorithme et suite de plus de 20 termes pour l'ε-algorithme). Cela a pour effet une augmentation de la place occupée par les résultats. Aussi, pour éviter d'accroître exagérément cet espace, n'avons nous pas affiché la totalité des résultats dans le cas des plus gros exemples mais seulement un extrait qui met en lumière le gain de précision substanciel apporté par la méthode.

6.1.2 Choix des exemples.

Pour illustrer la mise en œuvre de l'ε-algorithme, on a systématiquement adopté des suites vérifiant une relation de récurrence linéaire d'ordre k fixé,. Cela garantit que les termes de le colonne d'indice 2k sont égaux à l'anti-limite [**] de la suite et nous procure donc un moyen de contrôle particulièrement aisé. Dans ce cadre deux types d'exemples ont été retenus: des exemples où l'on rencontre des blocs totalement singuliers et des exemples où l'on rencontre des blocs quasi-singuliers. Les premiers mettent en lumière la fiabilité de l'algorithme alors que les seconds illustrent sa stabilité numérique.
Dans le cas du ρ-algorithme, il est moins facile de construire des exemples singuliers ou quasi-singuliers dont la solution soit connue. On trouvera certes des exemples qui mettent en relief la complexité de la structure des blocs singuliers qu'on peut rencontrer lors de la mise en œuvre d'un tel algorithme, mais le contrôle de la qualité des résultats ne sera plus assuré par la comparaison de la solution calculée avec la solution exacte.

6.1.3 Contrôle de la qualité des résultats.

Outre la comparaison avec la solution exacte qui est disponible dans les exemples retenus pour la mise en œuvre de l'ε-algorithme, on trouvera deux façons distinctes de contrôler la qualité des résultats. Le premier est tout simplement la consultation de la précision de chaque résultat estimée selon la méthode de perturbation qui a été retenue. On peut toutefois objecter qu'il n'est pas très sérieux de vérifier la qualité d'une méthode en s'appuyant sur un outil qui joue un rôle actif dans la mise en œuvre de cette méthode: on ne peut espérer détecter que d'éventuelles déficiences dont il ne serait pas responsables. Aussi ai-je introduit un contrôle tout à fait indépendant en exploitant une fois encore la propriété d'invariance anallagmatique des algorithmes de losange. J'ai tout simplement transformé les données (c'est à dire les deux colonnes d'indice -2 et 0) au moyen d'une inversion de puissance unité et dont le pôle a été choisi de façon à minimiser l'incertitude entachant les données modifiées. A cette fin, on a choisi comme pôle le point le plus éloigné de l'ensemble des données initiales (au sens de la distance riemannienne, et y compris la valeur ∞ de la colonne d'indice -2). L'application de notre algorithme à ces données transformées donne dans les colonnes dont l'indice est pair des valeurs qui sont les transformées par cette même transformation anallagmatique des valeurs obtenue par l'application directe de l'algorithme. Bien entendu, cette disposition qui double le temps de traitement ne doit pas être introduite de façon systématique dans la mise en œuvre des algorithmes de losange. Son seul objet est ici une vérification expérimentale de la qualité des résultats obtenus. Avant de clore cette question du contrôle de la qualité des résultats, faisons encore trois remarques:
1° La connaissance des résultats ainsi transformés s'accompagne encore d'une estimation de l'erreur entachant ces résultats transformés; pour chaque élément d'une colonne d'indice pair, on dis-

pose alors de trois estimations distinctes de l'erreur affectant ce résultat: l'estimation de l'erreur dans le tableau initial par la méthode de perturbation, l'estimation de l'erreur dans le tableau obtenu en appliquant la transformation inverse au tableau construit à partir des données transformées, l'écart entre les valeurs de ces deux tableaux. Ceci nous permettra de vérifier la cohérence des trois estimations, cohérence qui pourra encore être confirmée (ou infirmée) par la connaissance des résultats exacts dans le cas de l'ε-algorithme.

2° Cette réinversion des colonnes d'indices pair peut être complétée par un calcul des valeurs des éléments des colonnes d'indice impair par une utilisation systématique de la règle du losange dans ces colonnes. Le lecteur intéressé est invité à se reporter au sous programme qui réalise cette opération.

3° La méthode de contrôle (coûteuse) qui consiste à inverser les deux colonnes initiales peut être réutilisée autant de fois qu'on le voudra à condition de toujours choisir un pôle qui garantisse une bonne précision de la suite transformée.

6.1.4 Description de la présentation.

A chaque exemple sont associés trois types de tableaux:

* Le tableau des données, dont la présentation diffère selon qu'il s'agit de l'ε-algorithme ou du ρ-algorithme.
* Le tableau des résultats proprement dits: on affiche tout ou partie des colonnes du tableau calculé avec l'algorithme. Pour l'ε-algorithme, l'extrait comporte toujours la colonne où doit se rencontrer la valeur constante de l'anti-limite; pour le ρ-algorithme, on s'est parfois contenté d'afficher les dernières colonnes. Dans tous les cas, une indexation des lignes et des colonnes permet d'identifier le résultat affiché.
* Dans la troisième sorte de tableau, on trouvera le nombre de chiffres décimaux corrects estimés ou constatés. Trois tableaux distincts peuvent ainsi être présentés: les deux premiers fournissent le nombre de chiffres corrects estimés pour chaque élément du tableau principal, l'un lors de l'application directe de l'algorithme et l'autre lors de la réinversion des résultata obtenus par l'application de l'algorithme aux données inversées (le pôle d'inversion étant choisi selon la méthode esquissée au paragraphe précédent). Ces deux tableaux sont parfois omis. Par contre, le dernier tableau qui affiche le nombre de chiffres communs aux deux résultats est toujours présent. Rappelons à cet égard que la notion de nombre de chiffres communs à la représentation de deux nombres diffère de celle qui est le plus souvent retenue dans la littérature en ce sens que nous la calculons en prenant l'opposé du logarithme en base décimale de la distance riemannienne de ces nombres.

6.2 Les résultats présentés.

6.2.0 Les six exemples.

Nous avons retenu quatre exemples pour l'ε-algorithme: dans chacun des cas la suite vérifie une relation de récurrence (d'ordre 4 pour les deux premières et d'ordre 7 pour les deux autres). En fait le premier et le troisième exemple ont été choisis de façon à présenter des blocs totalement singuliers alors que le second et le quatrième exemple ont été respectivement obtenus en perturbant les données des exemples 1 et 3, ce qui conduit à des blocs presque singuliers. Si l'on ne craint pas de trop schématiser, on peut dire que les exemples 1 et 3 permettent de vérifier la fiabilité de l'algorithme alors que les exemples 2 et 4 permettent plutôt de contrôler sa stabilité numérique.

Pour le ρ-algorithme, trois exemples ont été retenus: le premier (de taille 8) et le second (de taille 18) correspondent à des difficultés mineures car les blocs singuliers, bien qu'étant non carrés sont tout de même bien distincts les uns des autres. Dans le dernier exemple (de taille 21), on rencontre un bloc singulier de très grande taille et de structure bien biscornue.

Nous laissons au lecteur le soin de contrôler la qualité des résultats: ils sont presque toujours excellents.

Précisons que les calculs ont été exécutés sur un Macintosh en utilisant la double précision dont nous disposons en Fortran MicroSoft, c'est à dire avec une mantisse de 52 chiffres binaires, ce qui équivaut à un peu moins de 16 chiffres décimaux.

6.2.1 Exemple 1

données (ε-algorithme, n=12)

indice	terme	erreur
0	0.200000000000000D+01	0.88818D-16
1	0.200000000000000D+01	0.88818D-16
2	0.200000000000000D+01	0.88818D-16
3	0.250000000000000D+01	0.61254D-16
4	0.300000000000000D+01	0.13878D-16
5	0.300000000000000D+01	0.44409D-16
6	0.300000000000000D+01	0.44409D-16
7	0.400000000000000D+01	0.13878D-16
8	0.500000000000000D+01	0.34161D-16
9	0.500000000000000D+01	0.34161D-16
10	0.500000000000000D+01	0.13878D-16
11	0.700000000000000D+01	0.17764D-16
12	0.900000000000000D+01	0.21663D-16

nombre de chiffres

	0	1	2	3	4	5	6	7	8	9	10	11
1	16.0											
		16.1										
2	16.1		00.0									
		16.1		00.0								
4	16.1		16.1		15.8							
		16.1		16.1		15.8						
6	16.2		15.0		15.9		15.1					
		16.2		15.0		15.9		15.1				
8	16.9		16.4		16.4		15.4		14.8			
		16.9		16.4		16.4		15.4		14.8		
10	16.4		00.0		16.9		14.6		14.4		00.0	
		16.4		00.0		16.9		14.6		14.4		OO.0
12	16.4		16.9		16.9		15.1		14.8		00.0	
		16.4		16.9		16.9		15.1		14.8		00.0
14	16.9		14.8		16.0		14.1		14.4		00.0	
		16.9		14.8		16.0		14.1		14.4		
16	16.5		16.9		16.5		15.1		14.5			
		16.5		16.9		16.5		15.1				
18	16.5		00.0		16.2		14.7					
		16.5		00.0		16.2						
20	16.9		16.5		16.2							
		16.9		16.5								
22	16.8		15.1									
		16.8										
24	16.7											

nombre de chiffres corrects du tableau initial

Annexe6: une version fiable et stable des algorithmes de losange.

	0	1	2	3	4	5	6	7	8	9	10	11
00	15.6											
		15.6										
02	16.1		00.0									
		16.1		00.0								
04	15.8		15.6		15.6							
		15.8		15.6		15.6						
06	15.9		14.0		15.3		13.8					
		15.9		14.0		15.3		13.8				
08	16.4		16.4		16.1		14.7		14.6			
		16.4		16.4		16.1		14.7		14.6		
10	15.9		00.0		16.4		14.7		15.3		14.6	
		15.9		00.0		16.4		14.7		15.3		14.6
12	15.9		15.9		16.4		15.4		14.9		00.0	
		15.9		15.9		16.4		15.4		14.9		00.0
14	16.1		14.7		15.7		14.1		12.8		00.0	
		16.1		14.7		15.7		14.1		12.8		
16	16.5		16.5		16.5		13.1		12.3			
		16.5		16.5		16.5		13.1				
18	16.5		00.0		16.9		12.9					
		16.5		00.0		16.9						
20	16.9		16.9		16.5							
		16.9		16.9								
22	16.3		15.2									
		16.3										
24	16.4											

nombre de chiffres corrects du tableau réinversé

	0	1	2	3	4	5	6	7	8	9	10	11
0	20.0											
		20.0										
2	20.0		00.0									
		20.0		00.0								
4	20.0		20.0		15.6							
		20.0		20.0		15.6						
6	20.0		16.6		15.3		16.6					
		20.0		16.6		15.3		16.6				
8	16.4		16.4		16.4		14.3		15.0			
		16.4		16.4		16.4		14.3		15.0		
10	20.0		00.0		16.4		14.1		14.5		00.0	
		20.0		00.0		16.4		14.1		14.5		00.0
12	20.0		20.0		20.0		15.0		14.5		00.0	
		20.0		20.0		20.0		15.0		14.5		00.0
14	20.0		16.6		15.4		16.6		12.6		00.0	
		20.0		16.6		15.4		16.6		12.6		
16	20.0		20.0		16.5		13.5		12.6			
		20.0		20.0		16.5		13.5				
18	20.0		00.0		16.5		16.6					
		20.0		00.0		16.5						
20	16.5		16.5		16.5							
		16.5		16.5								
22	20.0		16.6									
		20.0										
24	20.0											

nombre de chiffres communs aux deux tableaux

6.2.2 Exemple 2

données (ε-algorithme, n=12)

indice	terme	erreur
00	0.100000000000000D+01	0.11102D-15
01	0.100000000000000D+01	0.11102D-15
02	0.100010000000000D+01	0.11101D-15
03	0.150000000000000D+01	0.68321D-16
04	0.200000000000000D+01	0.13878D-16
05	0.200000000000000D+01	0.88818D-16
06	0.200020000000000D+01	0.88804D-16
07	0.300000000000000D+01	0.13878D-16
08	0.400000000000000D+01	0.52246D-16
09	0.400000000000000D+01	0.52246D-16
10	0.400040000000000D+01	0.13878D-16
11	0.600000000000000D+01	0.24005D-16
12	0.800000000000000D+01	0.27329D-16

tableau initial corrigé

	0	1	2	3	4	5	6	7	8	9
00	0.10000D+01									
		0.51923D+34								
02	0.10000D+01		0.10000D+01							
		0.10000D+01	-0.49970D+08							
04	0.10001D+01		0.10000D+01		0.10000D+01					
		0.20004D+01		0.20000D+01		0.40008D+01				
06	0.15000D+01		-0.24980D+04		0.14998D+01		-0.49864D+03			
		0.20000D+01		0.20004D+01		0.39988D+01		0.40008D+01		
08	0.20000D+01		0.20000D+01		0.20002D+01		0.99839D+00		0.19259D-33	
		0.51923D+34		0.50020D+04		0.30006D+01		0.29992D+01		0.51923D+34
10	0.20000D+01		0.20000D+01		0.20000D+01		-0.71232D+03		0.19259D-33	
		0.50000D+04		-0.24985D+08		0.29992D+01		0.30006D+01		0.51923D+34
12	0.20002D+01		0.20000D+01		0.20000D+01		0.99879D+00		0.19259D-33	
		0.10002D+01		0.10000D+01		0.20004D+01		0.19994D+01		0.51923D+34
14	0.30000D+01		-0.49960D+04		0.29996D+01		-0.99728D+03		0.19259D-33	
		0.10000D+01		0.10002D+01		0.19994D+01		0.20004D+01		0.51923D+34
16	0.40000D+01		0.40000D+01		0.40004D+01		0.19968D+01		0.19259D-33	
		0.51923D+34		0.25010D+04		0.15003D+01		0.14996D+01		
18	0.40000D+01		0.40000D+01		0.40000D+01		-0.14246D+04			
		0.25000D+04		-0.12492D+08		0.14996D+01				
20	0.40004D+01		0.40000D+01		0.40000D+01					
		0.50010D+00		0.50000D+00						
22	0.60000D+01		-0.99920D+04							
		0.50000D+00								
24	0.800000D+01									

Annexe6: une version fiable et stable des algorithmes de losange.

nombre de chiffres

	0	1	2	3	4	5	6	7	8	9	10	11	12
00	16.0												
		16.0											
02	16.0		16.0										
		16.0		16.0									
04	16.0		16.0		15.7								
		16.0		16.0		15.7							
06	16.2		14.7		15.6		14.2						
		16.2		14.7		15.6		14.2					
08	16.9		16.1		15.8		15.7		14.1				
		16.9		16.1		15.8		15.7		14.1			
10	16.1		16.1		16.1		14.4		14.0		00.0		
		16.1		16.1		16.1		14.4		14.0		00.0	
12	16.1		16.4		15.8		15.1		14.1		00.0		00.0
		16.1		16.4		15.8		15.1		14.1		00.0	
14	16.9		15.1		15.7		14.5		14.0		00.0		
		16.9		15.1		15.7		14.5		14.0			
16	16.3		16.3		16.3		14.8		13.9				
		16.3		16.3		16.3		14.8					
18	16.3		16.3		16.6		14.4						
		16.3		16.3		16.6							
20	16.9		16.3		16.3								
		16.9		16.3									
22	16.6		15.1										
		16.6											
24	16.6												

nombre de chiffres corrects du tableau direct

	0	1	2	3	4	5	6	7	8	9	10	11	12
0	15.7												
		15.7											
2	15.5		15.5										
		15.5		15.5									
4	15.5		15.3		15.5								
		15.5		15.3		15.5							
6	15.9		14.2		15.7		14.1						
		15.9		14.2		15.7		14.1					
8	16.1		16.1		15.6		14.7		14.0				
		16.1		16.1		15.6		14.7		14.0			
10	16.1		16.1		16.1		14.4		13.9		14.4		
		16.1		16.1		16.1		14.4		13.9		14.4	
12	16.1		15.7		15.8		14.4		14.0		13.7		00.0
		16.1		15.7		15.8		14.4		14.0		13.7	
14	16.1		14.5		15.4		14.3		13.2		00.0		
		16.1		14.5		15.4		14.3		13.2			
16	16.0		16.0		15.8		13.9		14.0				
		16.0		16.0		15.8		13.9					
18	16.3		16.1		16.6		13.4						
		16.3		16.1		16.6							
20	16.0		16.3		16.0								
		16.0		16.3									
22	15.9		14.5										
		15.9											
24	16.0												

nombre de chiffres corrects du tableau réinversé

	0	1	2	3	4	5	6	7	8	9	10	11	12
0	16.0												
		16.0											
2	15.7		15.7										
		15.7		15.7									
4	16.0		20.0		20.0								
		16.0		20.0		20.0							
6	16.2		14.6		15.2		14.0						
		16.2		14.6		15.2		14.0					
8	20.0		20.0		16.1		14.6		16.6				
		20.0		20.0		16.1		14.6		16.6			
10	16.1		20.0		16.1		14.2		16.6		00.0		
		16.1		20.0		16.1		14.2		16.6		00.0	
12	16.1		16.4		20.0		14.5		16.6		00.0		00.0
		16.1		16.4		20.0		14.5		16.6		00.0	
14	16.1		15.2		15.7		14.4		16.6		00.0		
		16.1		15.2		15.7		14.4		16.6			
16	16.3		16.3		20.0		15.7		16.6				
		16.3		16.3		20.0		15.7					
18	16.3		16.3		16.3		14.1						
		16.3		16.3		16.3							
20	16.0		16.1		16.3								
		16.0		16.1									
22	16.0		14.6										
		16.0											
24	16.1												

nombre de chiffres communs aux deux tableaux

6.2.3 Exemple 3.

données ε-algorithme, n=20

indice	terme	erreur
0	0.100000000000000D+01	0.11102D-15
1	0.100000000000000D+01	0.11102D-15
2	0.100000000000000D+01	0.13878D-16
3	0.100000000000000D+01	0.11102D-15
4	0.125000000000000D+01	0.86652D-16
5	0.150000000000000D+01	0.68321D-16
6	0.175000000000000D+01	0.54657D-16
7	0.200000000000000D+01	0.88818D-16
8	0.200000000000000D+01	0.88818D-16
9	0.200000000000000D+01	0.88818D-16
10	0.200000000000000D+01	0.88818D-16
11	0.250000000000000D+01	0.61254D-16
12	0.300000000000000D+01	0.44409D-16
13	0.350000000000000D+01	0.33516D-16
14	0.400000000000000D+01	0.52246D-16
15	0.400000000000000D+01	0.52246D-16
16	0.400000000000000D+01	0.52246D-16
17	0.400000000000000D+01	0.52246D-16
18	0.500000000000000D+01	0.13878D-16
19	0.600000000000000D+01	0.24005D-16
20	0.700000000000000D+01	0.17764D-16

tableau initial corrigé

```
      0      1      2      3      4      5      6      7      8      9     10     11
 0  0.10D+01
          0.519D+34
 2  0.10D+01   -0.51D+34
          0.51D+34   -0.51D+34
 4  0.10D+01   -0.51D+34   -0.51D+34
          0.51D+34    0.51D+34    0.51D+34
 6  0.10D+01    0.10D+01    0.10D+01    0.10D+01
          0.40D+01    0.40D+01    0.40D+01    0.40D+01
 8  0.12D+01    0.51D+34    0.51D+34    0.51D+34    0.12D+01
          0.40D+01   -0.51D+34   -0.51D+34    0.40D+01    0.80D+01
10  0.15D+01    0.51D+34   -0.51D+34    0.51D+34    0.15D+01    0.51D+34
          0.40D+01   -0.51D+34   -0.51D+34    0.40D+01    0.80D+01    0.80D+01
12  0.17D+01    0.51D+34    0.51D+34    0.51D+34    0.17D+01    0.13D+01
          0.40D+01    0.40D+01    0.40D+01    0.40D+01    0.53D+01    0.57D+01
14  0.20D+01    0.20D+01    0.20D+01    0.20D+01    0.25D+01    0.40D+01
          0.51D+34    0.51D+34    0.51D+34    0.60D+01    0.60D+01    0.60D+01
16  0.20D+01   -0.51D+34   -0.51D+34    0.20D+01    0.51D+34    0.51D+34
          0.51D+34   -0.51D+34    0.51D+34    0.60D+01   -0.51D+34    0.60D+01
18  0.20D+01   -0.51D+34   -0.51D+34    0.20D+01    0.51D+34    0.51D+34
          0.51D+34    0.51D+34    0.51D+34    0.60D+01    0.60D+01    0.60D+01
20  0.20D+01    0.20D+01    0.20D+01    0.20D+01    0.17D+01    0.17D+01
          0.20D+01    0.20D+01    0.20D+01    0.20D+01    0.33D+01    0.37D+01
22  0.25D+01    0.51D+34    0.51D+34    0.51D+34    0.25D+01    0.40D+01
          0.20D+01   -0.51D+34   -0.51D+34    0.20D+01    0.40D+01    0.40D+01
24  0.30D+01    0.51D+34   -0.51D+34    0.51D+34    0.30D+01    0.51D+34
          0.20D+01   -0.51D+34   -0.51D+34    0.20D+01    0.40D+01    0.40D+01
26  0.35D+01    0.51D+34    0.51D+34    0.51D+34    0.35D+01    0.27D+01
          0.20D+01    0.20D+01    0.20D+01    0.20D+01    0.26D+01    0.28D+01
28  0.40D+01    0.40D+01    0.40D+01    0.40D+01    0.50D+01    0.80D+01
          0.51D+34    0.51D+34    0.51D+34    0.30D+01    0.30D+01    0.30D+01
30  0.40D+01   -0.51D+34   -0.51D+34    0.40D+01    0.51D+34    0.51D+34
          0.51D+34   -0.51D+34    0.51D+34    0.30D+01   -0.51D+34
32  0.40D+01   -0.51D+34   -0.51D+34    0.40D+01    0.51D+34
          0.51D+34    0.51D+34    0.51D+34    0.30D+01
34  0.40D+01    0.40D+01    0.40D+01    0.40D+01
          0.10D+01    0.10D+01    0.10D+01
36  0.50D+01    0.51D+34    0.51D+34
          0.10D+01   -0.51D+34
38   0.60D+01    0.51D+34
          0.10D+01
40  0.70D+01
```

```
      12     13     14     15     16     17     18     19     20
12  0.93D+00
          0.58D+01
14  0.75D+01    0.19D-33
          0.57D+01    0.51D+34
16  0.37D+01    0.19D-33   -0.51D+34
          0.54D+01    0.51D+34   -0.51D+34
18  0.18D+01    0.19D-33   -0.51D+34   -0.51D+34
          0.49D+01    0.51D+34   -0.51D+34   -0.51D+34
20  0.93D+00    0.19D-33   -0.51D+34   -0.51D+34   -0.51D+34
          0.38D+01    0.51D+34   -0.51D+34   -0.51D+34
22  0.75D+01    0.19D-33   -0.51D+34   -0.51D+34
          0.37D+01    0.51D+34   -0.51D+34
24  0.37D+01    0.19D-33   -0.51D+34
          0.34D+01    0.51D+34
26  0.18D+01    0.19D-33
          0.29D+01
28  0.15D+02
```

nombre de chiffres corrects du tableau corrigé

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	**14**	15	16	17
0	16.0																	
		16.0																
2	16.0		.0															
		16.0		.0														
4	16.9		.0		.0													
		16.9		.0		.0												
6	16.0		16.0		16.0		16.9											
		16.0		16.0		16.0		16.9										
8	16.1		14.1		14.1		14.1		15.1									
		16.1		14.1		14.1		14.1		15.1								
10	16.2		14.1		.0		14.4		14.8		13.3							
		16.2		14.1		.0		14.4		14.8		13.3						
12	16.3		14.0		14.1		14.4		14.8		13.7		12.4					
		16.3		14.0		14.1		14.4		14.8		13.7		12.4				
14	16.1		16.9		16.1		16.1		15.4		14.5		14.7		13.8			
		16.1		16.9		16.1		16.1		15.4		14.5		14.7		13.8		
16	16.1		.0		.0		16.1		13.9		13.8		14.8		13.1		.0	
		16.1		.0		.0		16.1		13.9		13.8		14.8		13.1		.0
18	16.1		.0		.0		16.1		13.7		14.0		15.0		13.3		.0	
		16.1		.0		.0		16.1		13.7		14.0		15.0		13.3		.0
20	16.1		16.1		16.1		16.1		15.8		15.1		14.8		14.0		.0	
		16.1		16.1		16.1		16.1		15.8		15.1		14.8		14.0		.0
22	16.2		14.7		14.5		14.4		15.4		14.9		14.2		13.3		.0	
		16.2		14.7		14.5		14.4		15.4		14.9		14.2		13.3		.0
24	16.4		14.5		.0		14.5		15.4		13.7		15.1		13.2		.0	
		16.4		14.5		.0		14.5		15.4		13.7		15.1		13.2		
26	16.5		14.4		14.4		14.7		15.4		14.6		14.4		13.8			
		16.5		14.4		14.4		14.7		15.4		14.6		14.4				
28	16.3		16.3		16.3		16.9		15.6		15.2		14.7					
		16.3		16.3		16.3		16.9		15.6		15.2						
30	16.3		.0		.0		16.3		14.1		14.1							
		16.3		.0		.0		16.3		14.1								
32	16.3		.0		.0		16.6		14.5									
		16.3		.0		.0		16.6										
34	16.3		16.3		16.3		16.1											
		16.3		16.3		16.3												
36	16.9		14.8		14.7													
		16.9		14.8														
38	16.6		14.6															
		16.6																
40	16.8																	

Annexe6: une version fiable et stable des algorithmes de losange.

nombre de chiffres corrects du tableau réinversé

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
0	16.9																	
		16.9																
2	15.5		.0															
		15.5		.0														
4	15.4		.0		.0													
		15.4		.0		.0												
6	15.5		15.4		15.2		15.2											
		15.5		15.4		15.2		15.2										
8	15.8		13.5		13.6		13.8		12.4									
		15.8		13.5		13.6		13.8		12.4								
10	15.9		13.6		.0		13.6		11.9		12.5							
		15.9		13.6		.0		13.6		11.9		12.5						
12	16.0		13.8		13.6		13.5		13.4		11.4		11.8					
		16.0		13.8		13.6		13.5		13.4		11.4		11.8				
14	16.1		16.1		15.6		15.8		14.6		13.4		12.9		11.7			
		16.1		16.1		15.6		15.8		14.6		13.4		12.9		11.7		
16	15.8		.0		.0		15.8		13.4		13.3		13.1		10.7		.0	
		15.8		.0		.0		15.8		13.4		13.3		13.1		10.7		.0
18	15.6		.0		.0		15.8		13.3		13.4		11.9		10.5		.0	
		15.6		.0		.0		15.8		13.3		13.4		11.9		10.5		.0
20	16.1		15.6		15.8		16.1		15.2		12.0		11.7		10.8		.0	
		16.1		15.6		15.8		16.1		15.2		12.0		11.7		10.8		.0
22	16.2		14.6		14.4		14.0		11.6		10.5		10.5		10.1		.0	
		16.2		14.6		14.4		14.0		11.6		10.5		10.5		10.1		.0
24	15.9		14.4		.0		14.4		11.6		9.6		10.3		10.3		.0	
		15.9		14.4		.0		14.4		11.6		9.6		10.3		10.3		
26	16.5		14.0		14.4		14.5		11.8		11.1		10.6		10.4			
		16.5		14.0		14.4		14.5		11.8		11.1		10.6				
28	15.7		15.8		16.0		15.8		11.6		10.9		11.7					
		15.7		15.8		16.0		15.8		11.6		10.9						
30	15.7		.0		.0		15.9		10.0		10.1							
		15.7		.0		.0		15.9		10.0								
32	15.8		.0		.0		15.6		10.1									
		15.8		.0		.0		15.6										
34	15.5		15.7		15.9		15.7											
		15.5		15.7		15.9												
36	16.2		14.8		14.0													
		16.2		14.8														
38	15.9		14.1															
		15.9																
40	15.7																	

nombre de chiffres communs aux deux tableaux

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
0	16.0																	
		16.0																
2	20.0		.0															
		20.0		.0														
4	15.8		.0		.0													
		15.8		.0		.0												
6	20.0		16.0		16.3		15.6											
		20.0		16.0		16.3		15.6										
8	20.0		16.6		16.6		16.6		12.5									
		20.0		16.6		16.6		16.6		12.5								
10	16.2		16.6		.0		16.6		11.9		16.6							
		16.2		16.6		.0		16.6		11.9		16.6						
12	16.3		16.6		16.6		16.6		13.2		11.4		11.9					
		16.3		16.6		16.6		16.6		13.2		11.4		11.9				
14	16.1		20.0		20.0		16.1		14.9		13.2		12.8		16.6			
		16.1		20.0		20.0		16.1		14.9		13.2		12.8		16.6		
16	16.1		.0		.0		20.0		16.6		16.6		14.3		16.6		.0	
		16.1		.0		.0		20.0		16.6		16.6		14.3		16.6		.0
18	20.0		.0		.0		20.0		16.6		16.6		12.3		16.6		.0	
		20.0		.0		.0		20.0		16.6		16.6		12.3		16.6		.0
20	20.0		16.1		16.1		20.0		15.6		12.2		11.8		16.6		.0	
		20.0		16.1		16.1		20.0		15.6		12.2		11.8		16.6		.0
22	16.2		16.6		16.6		16.6		12.2		11.0		10.6		16.6		.0	
		16.2		16.6		16.6		16.6		12.2		11.0		10.6		16.6		.0
24	15.9		16.6		.0		16.6		11.6		16.6		10.7		16.6		.0	
		15.9		16.6		.0		16.6		11.6		16.6		10.7		16.6		
26	16.5		16.6		16.6		16.6		12.4		11.1		12.1		16.6			
		16.5		16.6		16.6		16.6		12.4		11.1		12.1				
28	16.3		16.3		16.3		16.0		11.7		10.8		11.1					
		16.3		16.3		16.3		16.0		11.7		10.8						
30	16.3		.0		.0		16.0		16.6		16.6							
		16.3		.0		.0		16.0		16.6								
32	16.3		.0		.0		16.0		16.6									
		16.3		.0		.0		16.0										
34	16.3		16.0		16.0		16.3											
		16.3		16.0		16.0												
36	16.5		16.6		16.6													
		16.5		16.6														
38	16.3		16.6															
		16.3																
40	16.1																	

6.2.4 Exemple 4.

données (ε-algorithme, n=20)

indice	terme	erreur
0	0.100000000000000D+01	0.11102D-15
1	0.100100000000000D+01	0.11091D-15
2	0.100200100000000D+01	0.13878D-16
3	0.100300200100000D+01	0.11069D-15
4	0.125000000000000D+01	0.86652D-16
5	0.150000100000000D+01	0.68321D-16
6	0.175000000000000D+01	0.54657D-16
7	0.200000000000000D+01	0.88818D-16
8	0.200200000000000D+01	0.88676D-16
9	0.200400200000000D+01	0.88534D-16
10	0.200600400200000D+01	0.88393D-16
11	0.250000000000000D+01	0.61254D-16
12	0.300000200000000D+01	0.44409D-16
13	0.350000000000000D+01	0.33516D-16
14	0.400000000000000D+01	0.52246D-16
15	0.400400000000000D+01	0.52148D-16
16	0.400800400000000D+01	0.52050D-16
17	0.401200800400000D+01	0.51952D-16
18	0.500000000000000D+01	0.13878D-16
19	0.600000400000000D+01	0.24005D-16
20	0.700000000000000D+01	0.17764D-16

tableau initial corrigé (colonnes 10 à 19)

	10	11	12	13	14	15	16	17	18	19
10	-0.189711D+04									
		0.795276D+01								
12	0.136776D+01		0.921786D+00							
		0.571048D+01		0.585142D+01						
14	0.413617D+01		0.801681D+01		0.192593D-33					
		0.596817D+01		0.572668D+01		0.519230D+34				
16	-0.153779D+06		0.387574D+01		0.192593D-33		-0.519230D+34			
		0.596817D+01		0.546867D+01		0.519230D+34		0.519230D+34		
18	-0.395405D+02		0.187376D+01		0.461575D-13		0.192593D-33		-0.519230D+34	
		0.599232D+01		0.493498D+01		-0.216649D+14		0.519230D+34		-0.519230D+34
20	0.136683D+01		0.927988D+00		0.192593D-33		0.192593D-33		-0.519230D+34	
		0.371360D+01		0.385738D+01		0.519230D+34		0.519230D+34		-0.519230D+34
22	0.407366D+01		0.788290D+01		0.192593D-33		-0.519230D+34		-0.519230D+34	
		0.397612D+01		0.373053D+01		0.519230D+34		-0.519230D+34		
24	-0.379423D+04		0.381112D+01		0.192593D-33		-0.519230D+34			
		0.397638D+01		0.346814D+01		0.519230D+34				
26	0.273552D+01		0.184357D+01		0.192593D-33					
		0.285524D+01		0.292571D+01						
28	0.827234D+01		0.160336D+02							
		0.298408D+01								
30	-0.307559D+06									

Annexe6: une version fiable et stable des algorithmes de losange.

nombre de chiffres corrects du tableau corrigé

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	**14**	15	16	17	18	19
0	16.0																			
		16.0																		
2	16.0		9.4																	
		16.0		9.4																
4	16.9		9.7		9.4															
		16.9		9.7		9.4														
6	16.0		16.9		15.7		15.7													
		16.0		16.9		15.7		15.7												
8	16.1		13.8		14.0		14.0		14.1											
		16.1		13.8		14.0		14.0		14.1										
10	16.2		14.0		14.0		14.2		14.2		12.9									
		16.2		14.0		14.0		14.2		14.2		12.9								
12	16.3		14.0		14.4		13.4		14.1		12.7		11.4							
		16.3		14.0		14.4		13.4		14.1		12.7		11.4						
14	16.1		16.1		15.6		15.8		15.3		13.9		14.2		13.2					
		16.1		16.1		15.6		15.8		15.3		13.9		14.2		13.2				
16	16.1		9.1		9.7		15.6		13.4		13.7		14.3		13.1		.0			
		16.1		9.1		9.7		15.6		13.4		13.7		14.3		13.1		.0		
18	16.1		9.7		9.2		16.1		13.6		13.7		14.5		13.8		13.2		.0	
		16.1		9.7		9.2		16.1		13.6		13.7		14.5		13.8		13.2		.0
20	16.1		15.6		15.8		15.9		15.8		15.1		14.6		14.3		13.8		.0	
		16.1		15.6		15.8		15.9		15.8		15.1		14.6		14.3		13.8		.0
22	16.2		14.3		14.2		14.3		15.4		14.3		14.3		13.3		.0		.0	
		16.2		14.3		14.2		14.3		15.4		14.3		14.3		13.3		.0		
24	16.4		14.2		14.4		14.4		15.3		13.5		14.1		13.2		.0			
		16.4		14.2		14.4		14.4		15.3		13.5		14.1		13.2				
26	16.5		14.1		14.7		14.4		15.8		15.0		14.1		13.5					
		16.5		14.1		14.7		14.4		15.8		15.0		14.1						
28	16.3		16.3		15.8		16.0		15.2		14.9		14.9							
		16.3		16.3		15.8		16.0		15.2		14.9								
30	16.3		9.1		9.8		15.7		13.7		14.0									
		16.3		9.1		9.8		15.7		13.7										
32	16.3		9.8		8.9		15.8		14.0											
		16.3		9.8		8.9		15.8												
34	16.3		15.8		15.7		15.7													
		16.3		15.8		15.7														
36	16.9		14.7		14.9															
		16.9		14.7																
38	16.6		14.8																	
		16.6																		
40	16.8																			

Annexe6: une version fiable et stable des algorithmes de losange.

nombre de chiffres corrects du tableau réinversé

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
0	15.5																			
		15.5																		
2	15.4		9.2																	
		15.4		9.2																
4	15.5		8.8		9.2															
		15.5		8.8		9.2														
6	15.7		15.2		15.4		15.4													
		15.7		15.2		15.4		15.4												
8	15.8		13.5		14.1		13.7		14.5											
		15.8		13.5		14.1		13.7		14.5										
10	15.9		14.0		13.7		14.1		14.1		11.8									
		15.9		14.0		13.7		14.1		14.1		11.8								
12	15.8		13.7		14.1		13.7		14.4		13.5		13.9							
		15.8		13.7		14.1		13.7		14.4		13.5		13.9						
14	16.1		16.1		15.6		15.6		15.2		14.6		14.2		13.3					
		16.1		16.1		15.6		15.6		15.2		14.6		14.2		13.3				
16	16.1		8.8		9.1		15.5		14.0		13.8		14.4		12.8		.0			
		16.1		8.8		9.1		15.5		14.0		13.8		14.4		12.8		.0		
18	16.1		9.1		8.8		16.1		13.8		14.0		14.4		13.2		.0		.0	
		16.1		9.1		8.8		16.1		13.8		14.0		14.4		13.2		.0		.0
20	15.6		15.8		16.1		15.7		15.6		15.0		14.4		13.7		.0		.0	
		15.6		15.8		16.1		15.7		15.6		15.0		14.4		13.7		.0		.0
22	16.2		14.5		14.4		14.2		14.6		13.5		13.1		10.3		.0		.0	
		16.2		14.5		14.4		14.2		14.6		13.5		13.1		10.3		.0		
24	16.4		14.3		14.2		14.4		14.4		12.3		12.0		10.5		.0			
		16.4		14.3		14.2		14.4		14.4		12.3		12.0		10.5				
26	16.2		14.2		14.4		14.6		14.5		11.8		11.8		11.4					
		16.2		14.2		14.4		14.6		14.5		11.8		11.8						
28	16.0		15.8		16.0		15.6		11.7		10.7		11.1							
		16.0		15.8		16.0		15.6		11.7		10.7								
30	15.8		8.5		9.4		15.8		9.8		9.8									
		15.8		8.5		9.4		15.8		9.8										
32	15.8		9.4		8.5		15.7		9.8											
		15.8		9.4		8.5		15.7												
34	15.7		15.8		15.5		15.8													
		15.7		15.8		15.5														
36	16.0		14.1		14.1															
		16.0		14.1																
38	15.8		14.1																	
		15.8																		
40	15.8																			

nombre de chiffres communs aux deux tableaux

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	**14**	15	16	17	18	19
0	16.0																			
		16.0																		
2	16.0		16.6																	
		16.0		16.6																
4	16.0		9.1		9.6															
		16.0		9.1		9.6														
6	20.0		15.4		15.7		16.3													
		20.0		15.4		15.7		16.3												
8	16.1		15.1		14.4		13.9		14.5											
		16.1		15.1		14.4		13.9		14.5										
10	16.2		14.3		13.9		14.6		15.0		12.6									
		16.2		14.3		13.9		14.6		15.0		12.6								
12	15.8		13.9		14.7		14.5		15.3		13.4		12.3							
		15.8		13.9		14.7		14.5		15.3		13.4		12.3						
14	16.1		20.0		20.0		15.6		15.4		14.9		14.7		16.6					
		16.1		20.0		20.0		15.6		15.4		14.9		14.7		16.6				
16	16.1		16.6		9.4		16.1		14.3		14.1		16.1		16.6		.0			
		16.1		16.6		9.4		16.1		14.3		14.1		16.1		16.6		.0		
18	20.0		9.4		9.4		16.1		14.1		13.8		14.5		13.3		.0		.0	
		20.0		9.4		9.4		16.1		14.1		13.8		14.5		13.3		.0		.0
20	15.8		15.8		16.1		16.4		15.8		15.6		14.8		16.6		.0		.0	
		15.8		15.8		16.1		16.4		15.8		15.6		14.8		16.6		.0		.0
22	16.2		14.2		14.2		14.5		15.2		14.2		14.4		16.6		.0		.0	
		16.2		14.2		14.2		14.5		15.2		14.2		14.4		16.6		.0		
24	20.0		14.3		14.7		14.3		15.4		13.7		12.3		16.6		.0			
		20.0		14.3		14.7		14.3		15.4		13.7		12.3		16.6				
26	16.5		14.3		14.6		14.1		15.5		12.0		12.3		16.6					
		16.5		14.3		14.6		14.1		15.5		12.0		12.3						
28	20.0		16.3		20.0		20.0		11.9		11.1		11.4							
		20.0		16.3		20.0		20.0		11.9		11.1								
30	20.0		16.6		9.5		15.7		10.1		10.2									
		20.0		16.6		9.5		15.7		10.1										
32	20.0		10.4		9.0		15.8		10.2											
		20.0		10.4		9.0		15.8												
34	16.0		16.0		16.3		15.7													
		16.0		16.0		16.3														
36	20.0		14.4		14.8															
		20.0		14.4																
38	16.6		14.8																	
		16.6																		
40	16.4																			

6.2.5 Exemple 5.

données (ρ-algorithme, n=8)

indice	abscisse	terme	err-coef	err-absc
0	0.19259299443D-33	-0.20000000000D+01	0.27756D-16	0.88818D-16
1	0.10000000000D+01	-0.10000000000D+01	0.11102D-15	0.11102D-15
2	0.20000000000D+01	0.19259299443D-33	0.13878D-16	0.27756D-16
3	0.30000000000D+01	0.19259299443D-33	0.44409D-16	0.27756D-16
4	0.40000000000D+01	0.19259299443D-33	0.52246D-16	0.27756D-16
5	0.50000000000D+01	0.10000000000D+01	0.13878D-16	0.11102D-15
6	0.60000000000D+01	0.19259299443D-33	0.24005D-16	0.27756D-16
7	0.70000000000D+01	-0.10000000000D+01	0.17764D-16	0.11102D-15
8	0.80000000000D+01	-0.20000000000D+01	0.27329D-16	0.88818D-16

tableau initial corrigé

	0	1	2	3	4	5	6	7	8
0	-0.20000D+01								
		0.10000D+01							
2	-0.10000D+01		0.51923D+34						
		0.10000D+01		0.10000D+01					
4	0.19259D-33		0.19259D-33		0.19259D-33				
		0.51923D+34		0.51923D+34		0.12000D+01			
6	0.19259D-33		-0.51923D+34		0.19259D-33		0.19259D-33		
		0.51923D+34		0.51923D+34		0.51923D+34		0.24285D+00	
8	0.19259D-33		0.19259D-33		-0.51923D+34		0.19259D-33		-0.77777D+01
		0.10000D+01		0.51923D+34		0.51923D+34		-0.78571D+00	
10	0.10000D+01		0.19259D-33		0.19259D-33		0.19259D-33		
		-0.10000D+01		-0.10000D+01		-0.10000D+01			
12	0.19259D-33		0.51923D+34		0.51923D+34				
		-0.10000D+01		-0.51923D+34					
14	-0.10000D+01		0.51923D+34						
		-0.10000D+01							
16	-0.20000D+01								

Annexe6: une version fiable et stable des algorithmes de losange.

tableau initial corrigé

	0	1	2	3	4	5	6	7	8
0	16.1								
		16.1							
2	16.0		15.7						
		16.0		15.7					
4	16.6		16.6		16.6				
		16.6		16.6		16.6			
6	16.6		.0		16.6		14.8		
		16.6		.0		16.6		14.8	
8	16.6		16.6		.0		15.6		15.3
		16.6		16.6		.0		15.6	
10	16.0		15.2		15.7		16.0		
		16.0		15.2		15.7			
12	16.6		15.4		15.2				
		16.6		15.4					
14	16.0		15.2						
		16.0							
16	16.1								

nombre de chiffres corrects estimé

tableau inverse réinversé

	0	1	2	3	4	5	6	7	8
0	15.9								
		15.9							
2	15.7		15.1						
		15.7		15.1					
4	16.3		15.7		15.7				
		16.3		15.7		15.7			
6	16.0		.0		15.7		15.2		
		16.0		.0		15.7		15.2	
8	15.7		15.7		.0		15.4		15.1
		15.7		15.7		.0		15.4	
10	15.4		15.7		16.0		15.2		
		15.4		15.7		16.0			
12	15.8		14.7		14.8				
		15.8		14.7					
14	15.7		15.3						
		15.7							
16	15.5								

nombre de chiffres communs aux deux tableaux

	0	1	2	3	4	5	6	7	8
0	15.9								
		15.9							
2	16.0		16.6						
		16.0		16.6					
4	15.8		16.6		16.6				
		15.8		16.6		16.6			
6	16.6		.0		16.6		16.6		
		16.6		.0		16.6		16.6	
8	16.6		16.6		.0		16.6		14.9
		16.6		16.6		.0		16.6	
10	16.0		16.6		16.6		16.6		
		16.0		16.6		16.6			
12	16.6		16.6		16.6				
		16.6		16.6					
14	16.0		16.6						
		16.0							
16	16.4								

6.2.6 Exemple 6.

données (ρ-algorithme, n=18)

indice	abscisse	terme	err-coef	err-absc
0	0.192592994438724D-33	-0.100000000000000D+01	0.27756D-16	0.11102D-15
1	0.100000000000000D+01	-0.100000000000000D+01	0.13878D-16	0.11102D-15
2	0.200000000000000D+01	0.192592994438724D-33	0.88818D-16	0.27756D-16
3	0.300000000000000D+01	-0.100000000000000D+01	0.44409D-16	0.11102D-15
4	0.400000000000000D+01	-0.100000000000000D+01	0.52246D-16	0.11102D-15
5	0.500000000000000D+01	0.192592994438724D-33	0.34161D-16	0.27756D-16
6	0.600000000000000D+01	0.192592994438724D-33	0.24005D-16	0.27756D-16
7	0.700000000000000D+01	0.100000000000000D+01	0.17764D-16	0.11102D-15
8	0.800000000000000D+01	0.192592994438724D-33	0.27329D-16	0.27756D-16
9	0.900000000000000D+01	0.192592994438724D-33	0.13878D-16	0.27756D-16
10	0.100000000000000D+02	0.100000000000000D+01	0.17588D-16	0.11102D-15
11	0.110000000000000D+02	0.100000000000000D+01	0.14560D-16	0.11102D-15
12	0.120000000000000D+02	0.200000000000000D+01	0.13878D-16	0.88818D-16
13	0.130000000000000D+02	0.100000000000000D+01	0.13878D-16	0.11102D-15
14	0.140000000000000D+02	0.100000000000000D+01	0.13878D-16	0.11102D-15
15	0.150000000000000D+02	0.200000000000000D+01	0.13878D-16	0.88818D-16
16	0.160000000000000D+02	0.200000000000000D+01	0.13878D-16	0.88818D-16
17	0.170000000000000D+02	0.300000000000000D+01	0.13878D-16	0.44409D-16
18	0.180000000000000D+02	0.200000000000000D+01	0.13878D-16	0.88818D-16

Annexe6: une version fiable et stable des algorithmes de losange.

tableau initial corrigé

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

 0 -0.100D+01
                0.519D+34
 2 -0.100D+01          -0.100D+01
                0.100D+01          0.519D+34
 4  0.192D-33          -0.100D+01          -0.519D+34
                -0.100D+01          0.519D+34          0.519D+34
 6 -0.100D+01          -0.100D+01          -0.100D+01          -0.100D+01
                0.519D+34          0.192D-33          0.500D+01          -0.132D+02
 8 -0.100D+01          -0.100D+01          0.192D-33          -0.138D+01          -0.961D+00
                0.100D+01          0.400D+01          0.666D+00          0.572D+01          0.150D+02
10  0.192D-33          0.192D-33          -0.150D+01          0.192D-33          0.192D-33          -0.103D+01
                0.519D+34          0.133D+01          0.466D+01          0.519D+34          0.537D+01          0.452D+01
12  0.192D-33          0.192D-33          0.192D-33          0.192D-33          0.192D-33          -0.139D+02
                0.100D+01          0.519D+34          0.519D+34          0.441D+01          0.466D+01          0.547D+01
14  0.100D+01          0.192D-33          -0.519D+34          0.192D-33          0.360D+02          -0.403D+00
                -0.100D+01          0.519D+34          0.519D+34          0.463D+01          0.438D+01          0.538D+01
16  0.192D-33          0.192D-33          0.192D-33          0.192D-33          -0.448D+00          0.106D+02
                0.519D+34          0.192D-33          0.500D+01          -0.132D+02          0.529D+01          0.414D+01
18  0.192D-33          0.192D-33          0.100D+01          -0.384D+00          0.381D-01          0.105D+01
                0.100D+01          0.400D+01          0.666D+00          0.572D+01          0.150D+02          0.499D+01
20  0.100D+01          0.100D+01          -0.500D+00          0.100D+01          0.100D+01          -0.309D-01
                0.519D+34          0.133D+01          0.466D+01          -0.311D+16          0.537D+01          0.452D+01
22  0.100D+01          0.100D+01          0.100D+01          0.100D+01          0.100D+01          -0.129D+02
                0.100D+01          0.519D+34          0.519D+34          0.441D+01          0.466D+01          0.547D+01
24  0.200D+01          0.100D+01          -0.519D+34          0.100D+01          0.370D+02          0.596D+00
                -0.100D+01          0.519D+34          0.519D+34          0.463D+01          0.438D+01          0.538D+01
26  0.100D+01          0.100D+01          0.100D+01          0.100D+01          0.551D+00          0.116D+02
                0.519D+34          0.192D-33          0.500D+01          -0.132D+02          0.529D+01
28  0.100D+01          0.100D+01          0.200D+01          0.615D+00          0.103D+01
                0.100D+01          0.400D+01          0.666D+00          0.572D+01
30  0.200D+01          0.200D+01          0.500D+00          0.200D+01
                0.519D+34          0.133D+01          0.466D+01
32  0.200D+01          0.200D+01          0.200D+01
                0.100D+01          0.519D+34
34  0.300D+01          0.200D+01
                -0.100D+01
36  0.200D+01

                12         13         14         15         16         17         18

12          -0.127D+01
                    0.537D+01
14          -0.135D+03          0.582D-01
                    0.548D+01          0.501D+01
16           0.981D+00          -0.322D+02          -0.209D+02
                    0.505D+01          0.643D+01          0.507D+01
18           0.151D+02          -0.213D+02          -0.334D+02          0.385D+02
                    0.467D+01          0.511D+01          0.532D+01
20          -0.257D+02          0.129D+02          0.477D+02
                    0.503D+01          0.557D+01
22          -0.273D+00          0.409D+02
                    0.537D+01
24          -0.134D+03
```

nombre de chiffres corrects du tableau corrigé

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
0	16.0																		
		16.0																	
2	16.0		16.0																
		16.0		16.0															
4	16.6		16.0		.0														
		16.6		16.0		.0													
6	16.0		16.3		15.8		15.7												
		16.0		16.3		15.8		15.7											
8	16.0		15.7		15.1		15.8		15.2										
		16.0		15.7		15.1		15.8		15.2									
10	16.6		16.6		15.5		15.1		15.4		15.3								
		16.6		16.6		15.5		15.1		15.4		15.3							
12	16.6		16.6		15.1		15.2		14.9		14.9		13.8						
		16.6		16.6		15.1		15.2		14.9		14.9		13.8					
14	16.0		15.1		.0		14.9		14.9		13.5		15.0		12.3				
		16.0		15.1		.0		14.9		14.9		13.5		15.0		12.3			
16	16.6		16.6		15.1		15.1		14.8		14.8		13.7		15.1		14.9		
		16.6		16.6		15.1		15.1		14.8		14.8		13.7		15.1		14.9	
18	16.6		16.6		15.5		14.8		15.0		15.0		14.9		15.3		15.0		14.6
		16.6		16.6		15.5		14.8		15.0		15.0		14.9		15.3		15.0	
20	16.0		15.5		14.8		15.6		15.1		14.8		15.1		14.5		14.8		
		16.0		15.5		14.8		15.6		15.1		14.8		15.1		14.5			
22	16.0		15.5		15.4		15.1		15.7		14.8		13.0		14.8				
		16.0		15.5		15.4		15.1		15.7		14.8		13.0					
24	16.1		15.6		.0		15.8		15.5		14.3		14.7						
		16.1		15.6		.0		15.8		15.5		14.3							
26	16.0		15.5		15.4		15.0		14.5		15.2								
		16.0		15.5		15.4		15.0		14.5									
28	16.0		15.7		15.8		14.6		14.9										
		16.0		15.7		15.8		14.6											
30	16.1		16.1		14.6		14.9												
		16.1		16.1		14.6													
32	16.1		15.8		16.1														
		16.1		15.8															
34	16.4		15.9																
		16.4																	
36	16.1																		

nombre de chiffres corrects du tableau réinversé

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
0	15.5																		
		15.5																	
2	15.5		15.3																
		15.5		15.3															
4	14.9		15.7		.0														
		14.9		15.7		.0													
6	15.3		14.9		15.1		14.8												
		15.3		14.9		15.1		14.8											
8	15.6		15.1		14.6		14.9		14.9										
		15.6		15.1		14.6		14.9		14.9									
10	14.8		14.7		14.9		14.2		14.2		14.7								
		14.8		14.7		14.9		14.2		14.2		14.7							
12	14.8		14.5		14.6		14.2		14.4		15.2		14.5						
		14.8		14.5		14.6		14.2		14.4		15.2		14.5					
14	14.6		14.7		.0		14.2		15.3		14.8		14.8		13.9				
		14.6		14.7		.0		14.2		15.3		14.8		14.8		13.9			
16	14.7		14.8		14.9		14.1		14.6		14.9		14.1		14.5		13.9		
		14.7		14.8		14.9		14.1		14.6		14.9		14.1		14.5		13.9	
18	14.9		14.7		15.2		14.5		13.9		14.0		15.2		14.0		14.4		13.8
		14.9		14.7		15.2		14.5		13.9		14.0		15.2		14.0		14.4	
20	14.6		14.3		14.0		14.2		14.1		13.9		14.0		14.6		13.9		
		14.6		14.3		14.0		14.2		14.1		13.9		14.0		14.6			
22	14.6		14.5		14.7		14.0		14.3		14.0		13.9		13.9				
		14.6		14.5		14.7		14.0		14.3		14.0		13.9					
24	14.7		14.5		.0		14.2		14.0		14.1		13.9						
		14.7		14.5		.0		14.2		14.0		14.1							
26	14.3		14.5		14.3		14.5		13.8		13.9								
		14.3		14.5		14.3		14.5		13.8									
28	14.6		14.1		14.4		13.9		14.3										
		14.6		14.1		14.4		13.9											
30	14.7		14.3		13.5		14.3												
		14.7		14.3		13.5													
32	14.7		14.1		14.2														
		14.7		14.1															
34	14.9		14.7																
		14.9																	
36	14.7																		

nombre de chiffres communs aux deux tableaux

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
0	20.0																		
		20.0																	
2	20.0		20.0																
		20.0		20.0															
4	16.6		16.0		.0														
		16.6		16.0		.0													
6	20.0		20.0		20.0		16.0												
		20.0		20.0		20.0		16.0											
8	20.0		15.8		16.6		15.6		15.3										
		20.0		15.8		16.6		15.6		15.3									
10	16.6		16.6		15.5		16.6		16.6		16.0								
		16.6		16.6		15.5		16.6		16.6		16.0							
12	16.6		16.6		16.6		16.6		16.6		15.0		14.5						
		16.6		16.6		16.6		16.6		16.6		15.0		14.5					
14	15.7		16.6		.0		16.6		14.9		14.4		15.2		12.5				
		15.7		16.6		.0		16.6		14.9		14.4		15.2		12.5			
16	16.6		16.6		16.6		16.6		15.2		14.5		13.7		14.8		14.3		
		16.6		16.6		16.6		16.6		15.2		14.5		13.7		14.8		14.3	
18	16.6		16.6		20.0		14.7		14.5		14.1		15.0		14.3		14.9		14.1
		16.6		16.6		20.0		14.7		14.5		14.1		15.0		14.3		14.9	
20	20.0		15.8		15.6		15.4		14.4		14.6		14.3		14.9		14.2		
		20.0		15.8		15.6		15.4		14.4		14.6		14.3		14.9			
22	15.8		14.6		14.7		14.4		15.6		14.3		13.8		14.3				
		15.8		14.6		14.7		14.4		15.6		14.3		13.8					
24	16.4		14.6		.0		15.3		14.4		14.0		14.3						
		16.4		14.6		.0		15.3		14.4		14.0							
26	14.7		14.6		14.7		14.7		14.0		14.4								
		14.7		14.6		14.7		14.7		14.0									
28	15.8		15.5		15.8		14.1		16.0										
		15.8		15.5		15.8		14.1											
30	16.1		14.7		15.3		14.9												
		16.1		14.7		15.3													
32	16.1		14.7		14.7														
		16.1		14.7															
34	14.7		14.8																
		14.7																	
36	16.1																		

6.2.7 Exemple 7.

données (ρ-algorithme, n=21)

indice	abscisse	terme	err-coef	err-absc
0	0.192592994438724D-33	0.200000000000000D+01	0.27756D-16	0.88818D-16
1	0.100000000000000D+01	0.192592994438724D-33	0.13878D-16	0.27756D-16
2	0.200000000000000D+01	-0.100000000000000D+01	0.88818D-16	0.11102D-15
3	0.300000000000000D+01	0.100000000000000D+01	0.44409D-16	0.11102D-15
4	0.400000000000000D+01	0.100000000000000D+01	0.52246D-16	0.11102D-15
5	0.500000000000000D+01	0.192592994438724D-33	0.34161D-16	0.27756D-16
6	0.600000000000000D+01	0.100000000000000D+01	0.24005D-16	0.11102D-15
7	0.700000000000000D+01	0.192592994438724D-33	0.17764D-16	0.27756D-16
8	0.800000000000000D+01	0.192592994438724D-33	0.27329D-16	0.27756D-16
9	0.900000000000000D+01	0.192592994438724D-33	0.13878D-16	0.27756D-16
10	0.100000000000000D+02	0.100000000000000D+01	0.17588D-16	0.11102D-15
11	0.110000000000000D+02	0.192592994438724D-33	0.14560D-16	0.27756D-16
12	0.120000000000000D+02	0.100000000000000D+01	0.13878D-16	0.11102D-15
13	0.130000000000000D+02	0.192592994438724D-33	0.13878D-16	0.27756D-16
14	0.140000000000000D+02	0.100000000000000D+01	0.13878D-16	0.11102D-15
15	0.150000000000000D+02	0.192592994438724D-33	0.13878D-16	0.27756D-16
16	0.160000000000000D+02	0.192592994438724D-33	0.13878D-16	0.27756D-16
17	0.170000000000000D+02	0.100000000000000D+01	0.13878D-16	0.11102D-15
18	0.180000000000000D+02	0.192592994438724D-33	0.13878D-16	0.27756D-16
19	0.190000000000000D+02	0.100000000000000D+01	0.13878D-16	0.11102D-15
20	0.200000000000000D+02	0.200000000000000D+01	0.13878D-16	0.88818D-16
21	0.210000000000000D+02	0.192592994438724D-33	0.13878D-16	0.27756D-16

Annexe6: une version fiable et stable des algorithmes de losange.

tableau initial corrigé

	0	1	2	3	4	5	6	7	8	9	10	11
0	0.200D+01											
		-0.500D+00										
2	0.192D-33		-0.400D+01									
		-0.100D+01		-0.307D+00								
4	-0.100D+01		0.333D+00		0.108D+01							
		0.500D+00		0.500D+01		-0.974D+00						
6	0.100D+01		0.100D+01		0.250D+00		0.107D+01					
		0.519D+34		-0.333D+00		0.633D+01		-0.180D+01				
8	0.100D+01		0.100D+01		0.100D+01		0.210D+00		-0.891D-01			
		-0.100D+01		0.519D+34		-0.126D+01		-0.285D+02		0.725D+02		
10	0.192D-33		0.100D+01		0.100D+01		-0.465D-01		0.192D-33		-0.102D+00	
		0.100D+01		-0.200D+01		-0.700D+01		0.143D+03		-0.246D+02		0.822D+02
12	0.100D+01		0.192D-33		0.192D-33		0.192D-33		-0.535D-01		0.192D-33	
		-0.100D+01		0.519D+34		0.519D+34		-0.600D+01		0.162D+03		-0.217D+02
14	0.192D-33		0.192D-33		-0.519D+34		0.192D-33		0.192D-33		-0.598D-01	
		0.519D+34		0.519D+34		0.519D+34		0.519D+34		-0.500D+01		0.178D+03
16	0.192D-33		-0.519D+34		-0.519D+34		-0.519D+34		0.192D-33		0.192D-33	
		0.519D+34		0.519D+34		0.519D+34		0.519D+34		0.519D+34		-0.400D+01
18	0.192D-33		0.192D-33		-0.519D+34		-0.519D+34		-0.519D+34		0.192D-33	
		0.100D+01		0.519D+34		0.519D+34		0.519D+34		0.519D+34		0.519D+34
20	0.100D+01		0.192D-33		0.192D-33		-0.519D+34		-0.519D+34		-0.519D+34	
		-0.100D+01		0.200D+01		0.519D+34		0.519D+34		0.519D+34		-0.519D+34
22	0.192D-33		0.100D+01		0.192D-33		0.192D-33		-0.519D+34		-0.519D+34	
		0.100D+01		-0.200D+01		0.300D+01		0.519D+34		-0.519D+34		-0.519D+34
24	0.100D+01		0.192D-33		0.100D+01		0.192D-33		-0.519D+34		-0.519D+34	
		-0.100D+01		0.200D+01		-0.300D+01		0.519D+34		0.519D+34		-0.519D+34
26	0.192D-33		0.100D+01		0.192D-33		0.192D-33		0.192D-33		-0.519D+34	
		0.100D+01		-0.200D+01		0.519D+34		-0.200D+01		0.519D+34		0.519D+34
28	0.100D+01		0.192D-33		0.192D-33		0.192D-33		0.192D-33		0.192D-33	
		-0.100D+01		0.519D+34		-0.100D+01		0.519D+34		-0.100D+01		0.375D+00
30	0.192D-33		0.192D-33		0.192D-33		0.192D-33		0.192D-33		0.800D+01	
		0.519D+34		0.192D-33		0.519D+34		0.192D-33		0.250D+00		-0.112D+01
32	0.192D-33		0.192D-33		0.192D-33		0.192D-33		0.360D+02		0.192D-33	
		0.100D+01		0.519D+34		0.100D+01		0.222D+00		-0.277D-01		
34	0.100D+01		0.192D-33		0.192D-33		-0.900D+01		0.192D-33			
		-0.100D+01		0.200D+01		0.333D+00		0.111D+01				
36	0.192D-33		0.100D+01		-0.300D+01		0.192D-33					
		0.100D+01		0.100D+01		0.233D+01						
38	0.100D+01		0.519D+34		0.750D+00							
		0.100D+01		0.100D+01								
40	0.200D+01		0.666D+00									
		-0.500D+00										
42	0.192D-33											

	12	13	14	15	16	17	18	19	20	21
12	-0.115D+00									
	0.908D+02									
14	0.192D-33	-0.127D+00								
	-0.192D+02	0.987D+02								
16	-0.656D-01	0.192D-33	0.192D-33							
	0.194D+03	0.519D+34	0.920D+02							
18	0.192D-33	0.192D-33	0.192D-33	0.192D-33						
	0.519D+34	0.181D+03	0.519D+34	0.867D+02						
20	-0.519D+34	0.192D-33	0.192D-33	0.192D-33	-0.201D+00					
	0.519D+34	0.519D+34	0.170D+03	-0.125D+02	0.916D+02					
22	-0.519D+34	-0.519D+34	0.192D-33	-0.103D+00	0.192D-33					
	-0.519D+34	0.519D+34	-0.291D+01	0.180D+03						
24	-0.519D+34	-0.519D+34	0.192D-33	0.192D-33						
	-0.519D+34	0.519D+34	0.519D+34							
26	-0.519D+34	-0.519D+34	-0.519D+34							
	0.519D+34	0.519D+34								
28	0.192D-33	-0.519D+34								
	0.519D+34									
30	0.192D-33									

Annexe6: une version fiable et stable des algorithmes de losange.

nombre de chiffres corrects du tableau corrigé

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
0	16.1																					
		16.1																				
2	16.6		15.4																			
		16.6		15.4																		
4	16.0		15.5		16.0																	
		16.0		15.5		16.0																
6	16.0		16.0		15.5		15.4															
		16.0		16.0		15.5		15.4														
8	16.0		15.5		15.7		15.6		15.5													
		16.0		15.5		15.7		15.6		15.5												
10	16.6		15.8		15.3		15.8		16.0		15.1											
		16.6		15.8		15.3		15.8		16.0		15.1										
12	16.0		15.4		15.8		16.1		15.7		15.1		15.1									
		16.0		15.4		15.8		16.1		15.7		15.1		15.1								
14	16.6		16.6		.0		15.7		15.1		15.6		14.7		15.0							
		16.6		16.6		.0		15.7		15.1		15.6		14.7		15.0						
16	16.6		.0		.0		.0		15.5		14.7		15.4		14.8		13.9					
		16.6		.0		.0		.0		15.5		14.7		15.4		14.8		13.9				
18	16.6		16.6		.0		.0		.0		15.2		14.8		13.9		15.0		14.2			
		16.6		16.6		.0		.0		.0		15.2		14.8		13.9		15.0		14.2		
20	16.0		15.3		15.8		.0		.0		.0		.0		15.0		14.2		15.1		14.7	
		16.0		15.3		15.8		.0		.0		.0		.0		15.0		14.2		15.1		14.7
22	16.6		15.7		15.1		15.4		.0		.0		.0		.0		15.1		15.1		13.6	
		16.6		15.7		15.1		15.4		.0		.0		.0		.0		15.1		15.1		
24	16.0		15.3		15.2		15.2		.0		.0		.0		.0		14.9		13.6			
		16.0		15.3		15.2		15.2		.0		.0		.0		.0		14.9				
26	16.6		15.4		15.1		16.1		15.1		.0		.0		.0		.0					
		16.6		15.4		15.1		16.1		15.1		.0		.0		.0						
28	16.0		15.4		16.1		15.0		14.9		15.1		14.8		.0							
		16.0		15.4		16.1		15.0		14.9		15.1		14.8								
30	16.6		16.6		15.5		15.0		14.9		14.8		13.5									
		16.6		16.6		15.5		15.0		14.9		14.8										
32	16.6		16.6		15.1		15.5		14.7		13.2											
		16.6		16.6		15.1		15.5		14.7												
34	16.0		15.2		15.7		14.5		13.9													
		16.0		15.2		15.7		14.5														
36	16.6		15.5		14.2		13.6															
		16.6		15.5		14.2																
38	16.0		14.4		13.9																	
		16.0		14.4																		
40	16.1		14.6																			
		16.1																				
42	16.6																					

Annexe6: une version fiable et stable des algorithmes de losange.

nombre de chiffres corrects du tableau réinversé

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
0	14.8																					
		14.8																				
2	15.1		14.9																			
		15.1		14.9																		
4	15.6		14.5		14.3																	
		15.6		14.5		14.3																
6	14.3		14.4		14.2		14.4															
		14.3		14.4		14.2		14.4														
8	14.8		14.1		14.3		14.2		14.1													
		14.8		14.1		14.3		14.2		14.1												
10	14.6		14.6		14.0		14.6		14.0		14.0											
		14.6		14.6		14.0		14.6		14.0		14.0										
12	14.5		15.4		14.4		13.9		14.4		14.1		14.1									
		14.5		15.4		14.4		13.9		14.4		14.1		14.1								
14	14.9		14.5		.0		14.2		14.0		14.4		14.0		13.9							
		14.9		14.5		.0		14.2		14.0		14.4		14.0		13.9						
16	14.7		.0		.0		.0		14.1		14.2		14.1		14.2		13.0					
		14.7		.0		.0		.0		14.1		14.2		14.1		14.2		13.0				
18	15.1		14.4		.0		.0		.0		14.4		14.1		13.1		13.8		13.0			
		15.1		14.4		.0		.0		.0		14.4		14.1		13.1		13.8		13.0		
20	14.7		15.1		14.4		.0		.0		.0		.0		14.2		13.0		13.9		14.1	
		14.7		15.1		14.4		.0		.0		.0		.0		14.2		13.0		13.9		14.1
22	14.9		14.6		14.9		14.6		.0		.0		.0		.0		14.0		14.1		13.7	
		14.9		14.6		14.9		14.6		.0		.0		.0		.0		14.0		14.1		
24	14.7		15.1		14.3		14.5		.0		.0		.0		.0		13.8		13.6			
		14.7		15.1		14.3		14.5		.0		.0		.0		.0		13.8				
26	14.6		14.3		14.6		14.6		14.9		.0		.0		.0		.0					
		14.6		14.3		14.6		14.6		14.9		.0		.0		.0						
28	14.5		14.8		14.4		14.4		16.6		14.3		14.2		.0							
		14.5		14.8		14.4		14.4		16.6		14.3		14.2								
30	15.1		14.6		14.8		14.7		14.1		13.9		14.2									
		15.1		14.6		14.8		14.7		14.1		13.9										
32	14.6		14.9		15.4		14.7		13.8		14.0											
		14.6		14.9		15.4		14.7		13.8												
34	14.5		14.9		14.6		13.8		14.2													
		14.5		14.9		14.6		13.8														
36	14.8		14.3		13.7		13.5															
		14.8		14.3		13.7																
38	14.3		13.9		13.8																	
		14.3		13.9																		
40	14.7		14.1																			
		14.7																				
42	14.6																					

Annexe6: une version fiable et stable des algorithmes de losange.

nombre de chiffres communs aux deux tableaux

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
0	20.0																						
		20.0																					
2	16.6		16.0																				
		16.6		16.0																			
4	16.3		14.9		15.0																		
		16.3		14.9		15.0																	
6	14.6		14.6		14.2		15.3																
		14.6		14.6		14.2		15.3															
8	15.7		14.6		20.0		14.2		14.7														
		15.7		14.6		20.0		14.2		14.7													
10	16.6		16.3		15.4		15.2		16.6		14.4												
		16.6		16.3		15.4		15.2		16.6		14.4											
12	15.7		16.6		16.6		16.6		15.8		16.6		14.3										
		15.7		16.6		16.6		16.6		15.8		16.6		14.3									
14	16.6		16.6		.0		16.6		16.6		15.4		16.6		14.3								
		16.6		16.6		.0		16.6		16.6		15.4		16.6		14.3							
16	16.6		.0		.0		.0		16.6		16.6		14.7		16.6		16.6						
		16.6		.0		.0		.0		16.6		16.6		14.7		16.6		16.6					
18	16.6		16.6		.0		.0		.0		16.6		16.6		16.6		16.6		16.6				
		16.6		16.6		.0		.0		.0		16.6		16.6		16.6		16.6		16.6			
20	20.0		16.6		16.6		.0		.0		.0		.0		16.6		16.6		16.6		14.1		
		20.0		16.6		16.6		.0		.0		.0		.0		16.6		16.6		16.6		14.1	
22	16.6		20.0		16.6		16.6		.0		.0		.0		.0		16.6		15.1		16.6		
		16.6		20.0		16.6		16.6		.0		.0		.0		.0		16.6		15.1			
24	15.7		16.6		14.6		16.6		.0		.0		.0		.0		16.6		16.6				
		15.7		16.6		14.6		16.6		.0		.0		.0		.0		16.6					
26	16.6		14.6		16.6		16.6		16.6		.0		.0		.0		.0						
		16.6		14.6		16.6		16.6		16.6		.0		.0		.0							
28	15.7		16.6		16.6		16.6		16.6		16.6		16.6		.0								
		15.7		16.6		16.6		16.6		16.6		16.6		16.6									
30	16.6		16.6		16.6		16.6		16.6		13.8		16.6										
		16.6		16.6		16.6		16.6		16.6		13.8											
32	16.6		16.6		16.6		16.6		13.8		16.6												
		16.6		16.6		16.6		16.6		13.8													
34	15.7		16.6		16.6		13.7		16.6														
		15.7		16.6		16.6		13.7															
36	16.6		14.6		13.7		16.6																
		16.6		14.6		13.7																	
38	14.6		16.6		16.1																		
		14.6		16.6																			
40	20.0		14.8																				
		20.0																					
42	16.6																						

7 Conclusion.

Les résultats précédents ne nécessitent aucun commentaire car ils illustrent clairement l'efficacité de la méthode proposée: rares sont les cas où une valeur extérieure à un bloc singulier est connue avec moins de 14 décimales; d'autre part, on constate la cohérence des diverses estimations des erreurs; enfin dans le cas de l'ε-algorithme, les valeurs théoriquement nulles s'identifient aisément. On peut donc affirmer que, sur le plan de la stabilité numérique, les résultats de cet algorithme répondent exactement à notre attente.

Bien entendu, toute médaille a son revers: cette solution n'est pas gratuite. Comme pour l'élaboration de tout logiciel, celle de ces algorithmes doit tenir compte de deux contraintes fondamentales: le coût opératoire et l'espace mobilisé. Certes, le coût des variantes algorithmiques proposées est sensiblement plus élevé que celui des algorithmes de losange initiaux, et la place mobilisée par la représentation des objets manipulés par ces variantes plus grande que le tableau unidimensionnel unique nécessité par la version initiale de l'ε-algorithme proposée par Wynn [24]. On peut estimer à quatre le facteur par lequel le coût est multiplié pour disposer d'une information fiable sur l'erreur, et à autant le facteur par lequel le coût est multiplié pour disposer des informations redondantes sur lequel l'algorithme est fondé. La complexité temporelle est alors pénalisée par un facteur dépassant dix et il est sera de même de la complexité spaciale. Néanmoins, il ne faut pas perdre de vue que, qu'elle soit temporelle ou spaciale, la complexité de l'algorithme présenté ici reste en $O(n^2)$ pour sa mise en œuvre sur un jeu de données de taille n. Le fait que les coûts opératoires ou spaciaux soient multipliés par un facteur supérieur à dix (mais qu'une programmation plus efficace permettrait certainement de diminuer quelque peu) est, nous semble-t-il, le juste prix qu'il faut payer pour pouvoir garantir la fiabilité et la stabilité d'un logiciel numérique.

7. Références

[1] Brezinski C.- Accélération de la convergence. Thèse d'Etat, Grenoble (1971).

[2] Brezinski C.- Généralisation des extrapolations polynomiales et rationnelles. RAIRO R1 (1972) 61-66.

[3] Brezinski C.- Conditions d'application et de convergence de procédés d'extrapolation. Numer. Math. 20 (1972) 64-79.

[4] Claessens G.- A useful identity for the rational Hermite interpolation table, Numer. Math. 26 (1978) pp 227-231.

[5] Cordellier F.- Une mise en œuvre numériquement stable de l'ε-algorithme vectoriel, Coll. Nat. d'Anal. Num. de Port-Bail (1976), Résumés.

[6] Cordellier F.- Particular rules for the vector ε-algorithm, Numer. Math. 27 (1977) pp 203-207.

[7] Cordellier F.- Démonstration algébrique de l' extension de l' identité de Wynn aux tables de Padé non normales. L.N.M. 765 (1979) 36-60.

[8] Cordellier F.- Erreurs numériques dans un espace compactifié. Coll. Nat. d'Anal. Num. de Bombanes (1982), Résumés.

[9] **Cordellier F.**- Règles pour l'ε-algorithme vectoriel. (Annexe 4)

[10] **Cordellier F.**- Utilisation de l'invariance homographique dans les algorithmes de losange, in Padé Approximation and its applications Bad Honnef 1983, H. Werner and H.J. Bûnger ed. Lect. Notes in Math. 1071 Springer-Verlag Berlin (1983) pp 62-94.

[11] **Cuyt A.**- Singular rules for the calculation of non-normal multivariate Padé approximants. J. Comp.Appl.Math. 14 (1986) 289-301.

[12] **Graves-Morris P.R. and Jenkins C.D.**- Degeneracies of generalised inverse, vector-valued approximants. A paraître dans Constr. Approx.

[13] **Hadamard J.**- Récents progrès de la géométrie anallagmatique. Œuvres Tome 2, Editions du CNRS (1968) 937-986.

[14] **Khéloufi R.**- Sur l'extension de la fiabilité des algorithmes de losange. Thèse de $3^{ème}$ cycle, Lille (1985).

[15] **La Porte M. et Vignes J.**- Error analysis computing. Proceedings of IFIP Congress, Stockholm (1974) 610-614.

[16] **Larkin F.M.**- Some techniques for rational interpolation, Computer J. 10 (1967) p.178-187.

[17] **Shanks D.**- Nonlinear transformations of divergent and slowly convergent sequences. J. Math.Phys. 34 (1955) 1-42.

[18] **Sterbenz P.H.**- Floating-point computation. Englewood cliffs N.J., Prentice Hall (1974).

[19] **Stoer J.**- Über zwei Algorithmen zur Interpolation mit Rationalen Funktionen, Numer. Math. 3 (1961) 285-305.

[20] **Thiele T.N.**- Interpolation Rechnung, Leipzig (1909).

[21] **Vignes J.**- Approche stochastique de l'analyse de propagation des erreurs d'arrondi et de données dans les algorithmes numériques. Ann. des Télécom. 41 (1986) 226-234.

[22] **Wynn P.**- On a device for computing the $e_m(S_n)$ transformation. M.T.A.C. 10 (1956) 91-96.

[23] **Wynn P.**- L'ε-algorithma e la tavola di Padé. Rend. di Mat. Roma 20 (1961) 403-408.

[24] **Wynn P.**- Acceleration techniques for iterated vector and matrix problems. Math. Comp. 16 (1962) 301-322.

[25] **Wynn P.**- Acceleration techniques in numerical analysis, with particular reference to problems in one independent variable. Proceedings IFIP Congress 1962, North Holland Pub. Co. (1963) 149-156.

[26] **Wynn P.**- Singular rules for certain nonlinear algorithms. B.I.T. 3 (1963) 175-195.(1963)

[26] **Wynn P.**- Upon systems of recursions which obtain among the quotients of the Padé table. Numer. Math. 8 (1966) 264-269.