

S 376  
1 977  
3

N° d'ordre : 644

55376  
1977  
3

**THESE**

présentée à

L'UNIVERSITE DES SCIENCES ET TECHNIQUES DE LILLE I

pour obtenir le titre de

**DOCTEUR DE SPECIALITE**

(Mathématiques appliquées)

par

**Didier PETIT**

**ETUDE DE CERTAINS PROCÉDÉS  
D'ACCÉLÉRATION DE LA CONVERGENCE  
EN ANALYSE NUMÉRIQUE**



Thèse soutenue le 17 Juin 1977, devant la Commission d'Examen

MM.	P.	BACCHUS	Président
	P.	HUARD	Examineur
	C.	BREZINSKI	Rapporteur

SCD LILLE 1



D 030 199919 8



DOYENS HONORAIRES de l'Ancienne Faculté des Sciences

MM. R. DEFRETIN, H. LEFEBVRE, M. PARREAU.

PROFESSEURS HONORAIRES des Anciennes Facultés de Droit  
et Sciences Economiques, des Sciences et des Lettres

M. ARNOULT, Mme BEAUJEU, MM. BROCHARD, CHAPPELON, CHAUDRON, CORDONNIER, CORSIN, DEHEUVELS  
DEHORS, DION, FAUVEL, FLEURY, P. GERMAIN, HEIM DE BALSAC, HOCQUETTE, KAMPE DE FERIET,  
KOGANOFF, LAMOTTE, LASSERRE, LELONG, Mme LELONG, MM. LHOMME, LIEBAERT, MARTINOT-LAGARDE,  
MAZET, MICHEL, NORMANT, PEREZ, ROIG, ROSEAU, ROUBINE, ROUELLE, SAVART, WATERLOT, WIEMAN,  
ZAMANSKI.

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

MM. R. DEFRETIN, M. PARREAU.

PRESIDENT DE L'UNIVERSITE  
DES SCIENCES ET TECHNIQUES DE LILLE

M. J. LOMBARD.

PROFESSEURS TITULAIRES

M. BACCHUS Pierre	Astronomie
M. BEAUFILS Jean-Pierre	Chimie Physique
M. BECART Maurice	Physique Atomique et Moléculaire
M. BILLARD Jean	Physique du Solide
M. BIAYS Pierre	Géographie
M. BONNEIAN Pierre	Chimie Appliquée
M. BONNOT Ernest	Biologie Végétale
M. BONTE Antoine	Géologie Appliquée
M. BOUGHON Pierre	Algèbre
M. BOURIQUET Robert	Biologie Végétale
M. CELET Paul	Géologie Générale
M. CONSTANT Eugène	Electronique
M. DECUYPER Marcel	Géométrie
M. DELATTRE Charles	Géologie Générale
M. DELHAYE Michel	Chimie Physique
M. DERCOURT Michel	Géologie Générale
M. DURCHON Maurice	Biologie Expérimentale
M. FAURE Robert	Mécanique
M. FOURET René	Physique du Solide
M. GABILLARD Robert	Electronique
M. GLACET Charles	Chimie Organique
M. GONTIER Gérard	Mécanique
M. GRUSON Laurent	Algèbre
M. GUILLAUME Jean	Microbiologie
M. HEUBEL Joseph	Chimie Minérale
M. LABLACHE-COMBIER Alain	Chimie Organique
M. LANSRAUX Guy	Physique Atomique et Moléculaire
M. LAVEINE Jean-Pierre	Paléontologie
M. LEBRUN André	Electronique
M. LEHMANN Daniel	Géométrie

Mme	LENOBLE Jacqueline	Physique Atomique et Moléculaire
M.	LINDER Robert	Biologie et Physiologie Végétales
M.	LOMBARD Jacques	Sociologie
M.	LOUCHEUX Claude	Chimie Physique
M.	LUCQUIN Michel	Chimie Physique
M.	MAILLET Pierre	Sciences Economiques
M.	MONTARIOL Frédéric	Chimie Appliquée
M.	MONTREUIL Jean	Biochimie
M.	PARREAU Michel	Analyse
M.	POUZET Pierre	Analyse Numérique
M.	PROUVOST Jean	Minéralogie
M.	SALMER Georges	Electronique
M.	SCHILTZ René	Physique Atomique et Moléculaire
Mme	SCHWARTZ Marie-Hélène	Géométrie
M.	SEGUIER Guy	Electrotechnique
M.	TILLIEU Jacques	Physique Théorique
M.	TRIDOT Gabriel	Chimie Appliquée
M.	VIDAL Pierre	Automatique
M.	VIVIER Emile	Biologie Cellulaire
M.	WERTHEIMER Raymond	Physique Atomique et Moléculaire
M.	ZEYTOUNIAN Radyadour	Mécanique

#### PROFESSEURS SANS CHAIRE

M.	BELLET Jean	Physique Atomique et Moléculaire
M.	BODARD Marcel	Biologie Végétale
M.	BOILLET Pierre	Physique Atomique et Moléculaire
M.	BOILLY Bénoni	Biologie Animale
M.	BRIODOUX Michel	Chimie Physique
M.	CAPURON Alfred	Biologie Animale
M.	CORTOIS Jean	Physique Nucléaire et Corpusculaire
M.	DEBOURSE Jean-Pierre	Gestion des entreprises
M.	DEPREZ Gilbert	Physique Théorique
M.	DEVRAINNE Pierre	Chimie Minérale
M.	GOUMAND Pierre	Chimie Physique
M.	GUILBAULT Pierre	Physiologie Animale
M.	LACOSTE Louis	Biologie Végétale
Mme	LEHMANN Josiane	Analyse
M.	LENTACKER Firmin	Géographie
M.	LOUAGE Francis	Electronique
Mle	MARQUET Simone	Probabilités
M.	MIGEON Michel	Chimie Physique
M.	MONTEL Marc	Physique du Solide
M.	PANET Marius	Electrotechnique
M.	RACZY Ladislas	Electronique
M.	ROUSSEAU Jean-Paul	Physiologie Animale
M.	SLIWA Henri	Chimie Organique

#### MAITRES DE CONFERENCES (et chargés d'Enseignement)

M.	ADAM Michel	Sciences Economiques
M.	ANTOINE Philippe	Analyse
M.	BART André	Biologie Animale
M.	BEGUIN Paul	Mécanique
M.	BROUCHE Rudolphe	Algèbre
M.	BONNELLE Jean-Pierre	Chimie
M.	BONNEMAIN Jean-Louis	Biologie Végétale
M.	BOSCO Denis	Probabilités
M.	BREZINSKI Claude	Analyse Numérique
M.	BRUYELLE Pierre	Géographie

M. CARREZ Christian	Informatique
M. CORDONNIER Vincent	Informatique
M. COQUERY Jean-Marie	Psycho-Physiologie
M <sup>le</sup> DACHARRY Monique	Géographie
M. DEBENEST Jean	Sciences Economiques
M. DEBRABANT Pierre	Géologie Appliquée
M. DE PARIS Jean-Claude	Mathématiques
M. DHAINAUT André	Biologie Animale
M. DELAUNAY Jean-Claude	Sciences Economiques
M. DERIEUX Jean-Claude	Microbiologie
M. DOUKHAN Jean-Claude	Physique du Solide
M. DUBOIS Henri	Physique
M. DYMENT Arthur	Mécanique
M. ESCAIG Bertrand	Physique du Solide
M <sup>e</sup> EVRARD Micheline	Chimie Appliquée
M. FONTAINE Jacques-Marie	Electronique
M. FOURNET Bernard	Biochimie
M. FORELICH Daniel	Chimie Physique
M. GAMBLIN André	Géographie
M. GOBLOT Rémi	Algèbre
M. GOSSELIN Gabriel	Sociologie
M. GRANELLE Jean-Jacques	Sciences Economiques
M. GUILLAUME Henri	Sciences Economiques
M. HECTOR Joseph	Géométrie
M. JACOB Gérard	Informatique
M. JOURNEL Gérard	Physique Atomique et Moléculaire
M <sup>le</sup> KOSMAN Yvette	Géométrie
M. KREMBEL Jean	Biochimie
M. LAURENT François	Automatique
M <sup>le</sup> LEGRAND Denise	Algèbre
M <sup>le</sup> LEGRAND Solange	Algèbre
M. LERDY Jean-Marie	Chimie Appliquée
M. LEROY Yves	Electronique
M. LHENAFF René	Géographie
M. LOCQUENEUX Robert	Physique Théorique
M. LOUCHET Pierre	Sciences de l'Education
M. MACKE Bruno	Physique
M. MAHIEU Jean-Marie	Physique Atomique et Moléculaire
M <sup>e</sup> N'GUYEN VAN CHI Régine	Géographie
M. MAIZIERES Christian	Automatique
M. MALAUSSENA Jean-Louis	Sciences Economiques
M. MESSELYN Jean	Physique Atomique et Moléculaire
M. MONTUELLE Bernard	Biologique Appliquée
M. NICOLE Jacques	Chimie Appliquée
M. PAQUET Jacques	Géologie Générale
M. PARSY Fernand	Mécanique
M. PECQUE Marcel	Chimie Physique
M. PERROT Pierre	Chimie Appliquée
M. PERTUZON Emile	Physiologie Animale
M. PONSOLLE Louis	Chimie Physique
M. POVY Lucien	Automatique
M. RICHARD Alain	Biologie
M. ROGALSKI Marc	Analyse
M. ROY Jean-Claude	Psycho-Physiologie
M. SIMON Michel	Sociologie
M. SOMME Jean	Géographie
M <sup>le</sup> SPIK Geneviève	Biochimie
M. STANKIEWICZ François	Sciences Economiques
M. STEEN Jean-Pierre	Informatique

M. THERY Pierre  
M. TOULOTTE Jean-Marc  
M. TREANTON Jean-René  
M. VANDORPE Bernard  
M. VILLETTE Michel  
M. WALLART Francis  
M. WERNIER Georges  
M. WATERLOT Michel  
Mme ZINN-JUSTIN Nicole

Electronique  
Automatique  
Sociologie  
Chimie Minérale  
Mécanique  
Chimie  
Informatique  
Géologie Générale  
Algèbre

*Je tiens à exprimer toute ma gratitude à Monsieur le Professeur BACCHUS qui me fait l'honneur de présider le jury de cette thèse.*

*Je remercie vivement Monsieur le Professeur HUARD pour l'intérêt qu'il a porté à ce travail en acceptant bien volontiers de le juger.*

*Que Monsieur le Professeur BREZINSKI dont les conseils, les suggestions et l'aide constante m'ont été d'un précieux secours, veuille trouver ici l'expression de ma plus vive reconnaissance.*

*Que Mademoiselle DESQUIENS, Mademoiselle DRIESSENS ainsi que Monsieur et Madame DEBOCK qui n'ont ménagé ni leur temps ni leur peine pour en assurer la réalisation matérielle, trouvent ici l'expression de mes plus sincères remerciements.*

*A ma femme,  
A mes parents.*

# PLAN

## INTRODUCTION

### I - RAPPELS ET DEFINITIONS

- I.1 COMPARAISONS DE SUITES
- I.2 DÉFINITIONS

### II ALGORITHME DE TYPE $\theta$

- II.1 FORMULATION GÉNÉRALE
- II.2 PROPRIÉTÉS
- II.3 ALGORITHME DE TYPE  $\theta_2$ 
  - II.3.1 Théorème de Convergence
  - II.3.2 Suites à convergence logarithmique
  - II.3.3 Application répétée des algorithmes de type  $\theta_2$
- II.4 ALGORITHME DE TYPE  $\theta$ . GÉNÉRALISATION AUX AUTRES COLONNES
- II.5 STABILITÉ NUMÉRIQUE ET PROPAGATION DES ERREURS

### III APPLICATIONS AUX ALGORITHMES CONNUS

- III.1 L' $\epsilon$ -ALGORITHME
- III.2 LE  $\epsilon$ -ALGORITHME
- III.3 GÉNÉRALISATION DE L' $\epsilon$ -ALGORITHME

### IV ETUDE DU PROCÉDÉ Q

- IV.1 NOTATIONS ET PRÉLIMINAIRES
- IV.2 ALGORITHME DE CALCUL
- IV.3 VISUALISATION DE L'ALGORITHME
- IV.4 THÉORÈMES DE CONVERGENCE
- IV.5 APPLICATIONS
  - IV.5.1 Intégrales impropres du premier ordre
  - IV.5.2 Relation avec l' $\epsilon$ -algorithme

### V APPLICATIONS ET RESULTATS NUMÉRIQUES

- V.1 LIMITES DE SUITES ET SOMMES DE SÉRIES
- V.2 APPLICATIONS AUX PROBLÈMES D'ANALYSE NUMÉRIQUE
  - a) Recherche des valeurs propres d'une matrice
  - b) Systèmes d'équations
  - c) Problèmes de quadrature numérique

## ERRATA

Page	Ligne	
7	5	lire "en une suite"
8	7	lire "auxiliaires"
10	-2	lire " $a_k^{(n)} = \Delta x_n \dots$ "
10	-1	lire "auxiliaire"
12	2	lire " $\theta_{2k}^{(n)}, \theta_{2k}^{(n+1)}, \theta_{2k}^{(n+2)} \dots$ "
14	3	lire "auxiliaire ..."
20	11	lire "auxiliaires ..."
23	-1	lire "appellerons $K_0 \dots$ "
26	14	lire "Log(3) = 1,09 ..."
27	9	lire " $\lim_{n \rightarrow \infty} \Delta S_{n+1} / \Delta S_n \dots$ "
27	-2	lire " $\lim_{n \rightarrow \infty} (\Delta S_{n+1} / \Delta S_n) = \dots$ "
42	-2	lire " $c_k^{(n)} = a_{2k+1}^{(n)} / a_{2k}^{(n)} \dots$ "
45	12	lire " $\Delta \theta_{2j+2}^{(n)} \dots$ "
47	13	lire " $\epsilon_{2n}^{(o)} \dots$ "
59	10	lire "anti-limite ..."
59	12	lire "d'accumulation ..."
68	-2	lire "cela nous donnons ..."
117	-4	lire "dans [44] ..."

## INTRODUCTION

Soit  $\{S_n\}$  une suite qui converge vers  $S$ .

Un problème que l'on rencontre fréquemment en analyse numérique est la détermination de  $S$  à partir d'un certain nombre de termes  $S_n$ . Ce problème peut être abordé de deux manières qui, à priori, semblent totalement différentes :

la première méthode consiste à transformer la suite  $S_n$  en une suite  $V_n$  qui converge "plus vite" vers la même limite  $S$ . On effectue cette transformation à l'aide d'un opérateur linéaire  $T$ . Les propriétés de convergence et d'"accélération de la convergence" sont directement liées à  $T$ . On appelle ces méthodes : procédés de sommation [1, 3, 66] .

Le second point de vue est celui de l'extrapolation tels les procédés de Richardson [47] et de Romberg, basés sur l'extrapolation polynomiale, qui de ce fait sont linéaires ou les  $\epsilon$  et  $\rho$ -algorithmes basés sur l'extrapolation par des sommes d'exponentielles ou par des fractions rationnelles qui donnent des procédés non linéaires. En réalité les procédés de sommation sont des cas particuliers de l'extrapolation par des sommes d'exponentielles. Nous étudierons ici certains de ces procédés d'extrapolations non linéaires.

L'algorithme d'extrapolation le plus puissant à l'heure actuelle est sans doute l' $\epsilon$ -algorithme. D'autre part, c'est pour cet algorithme que l'on possède le plus de résultats théoriques.

Cet algorithme a été introduit par Wynn [55] ; il permet de mettre en oeuvre la généralisation du procédé bien connu d'"accélération de la convergence" qu'est le  $\Delta^2$  d'Aitken.

Ce procédé a été imaginé par A. C. Aitken [1] en 1926. Il fut repris en 1959 par Schmidt [49], en 1952 par Lubkin [39] puis généralisé en 1955 par Shanks [50].

En 1956, Wynn a proposé une méthode récursive pour effectuer cette transformation de Shanks : c'est l' $\epsilon$ -algorithme. Il s'est depuis presque entièrement consacré à son étude et l'a notamment relié à la théorie des fractions continues, à la table de Padé et à certaines équations aux dérivées partielles. Bauer [2] en 1957 a mis en évidence, à l'aide des fractions continues de Stieljes, les relations qui existent entre l' $\epsilon$ -algorithme, l'algorithme Q.D de Rutishauser [48] et une classe plus générale d'algorithmes : les algorithmes en losange. Brezinski [3, 5, 18, 6, 7, 8, 9, 10, 11, 66] a étudié certains de ces procédés non linéaires :  $\epsilon$ ,  $\rho$  algorithme, procédé d'Overholt ; et en a énoncé de nouveaux : les deux généralisations de l' $\epsilon$ -algorithme, l'extension du  $\rho$ -algorithme, le  $\theta$ -algorithme et les procédés p et q. Depuis divers auteurs comme Clark [17, 18], Gray, Atchison, Adams et Schucany [29, 30, 31, 32, 33, 34] ont étudié la convergence et l'application au calcul d'intégrales impropres de transformations dérivées du procédé classique d'Aitken.

Nous avons noté que divers de ces algorithmes de transformations non linéaires ont une formulation commune.

Nous appelons cette formulation:algorithmes de type  $\theta$  où chaque algorithme sera caractérisé par des suites  $b_n$  et  $a_k^{(n)}$ . Ainsi le but de ce travail est l'étude systématique des algorithmes de type  $\theta$ .

Avant d'étudier ces méthodes nous avons besoin d'un certain nombre de concepts et de définitions qui forment le premier chapitre. Le deuxième chapitre traite des algorithmes de type  $\theta$  dans toute leur généralité. Après avoir énoncé quelques propriétés, nous consacrons une partie du chapitre à l'étude des algorithmes de type  $\theta_2$  ; en effet les algorithmes de type  $\theta$  ne sont qu'un cas particulier des algorithmes de type  $\theta_2$  itérés. Nous séparons, dans cette étude, les suites à convergence logarithmique des autres suites et nous donnons dans chaque cas des conditions de conservations de la limite

et d'"accélération de la convergence". Ces conditions sont ensuite généralisées aux autres colonnes du tableau 0.

Ce chapitre se termine par une étude de la stabilité numérique et de la propagation des erreurs.

Dans le troisième chapitre nous appliquons les résultats du chapitre précédent aux algorithmes déjà connus :  $\epsilon$ -algorithme,  $\rho$ -algorithme, première et deuxième généralisation de l' $\epsilon$ -algorithme. Nous retrouvons ainsi des résultats connus mais surtout de nouveaux résultats : condition de convergence et d'"accélération de convergence" pour l' $\epsilon$ -algorithme et les autres algorithmes pour lesquels nous n'avions que peu de résultats jusqu'à présent.

Le quatrième chapitre traite du procédé  $q$  introduit par BREZINSKI. Les éléments successifs de ce procédé se mettent sous la forme d'un rapport de deux déterminants et nous mettons en oeuvre un algorithme récursif qui permet le calcul de ce rapport. Cet algorithme nous permet d'obtenir certains résultats de convergence et d'"accélération de la convergence" pour le procédé  $q$ . Ce procédé est relié à l' $\epsilon$ -algorithme et nous utilisons cette relation pour donner un résultat important de convergence pour l' $\epsilon$ -algorithme.

Enfin le cinquième chapitre expose les applications et les résultats numériques obtenus.

## CHAPITRE I

### RAPPELS ET DÉFINITIONS

#### I.] COMPARAISON DE SUITES.

Pour ces définitions nous utiliserons systématiquement les relations de comparaison. Leur intérêt réside dans le fait qu'elles ne supposent pas l'existence de limites quand  $n$  tend vers l'infini.

Soient  $\{u_n\}$  et  $\{v_n\}$  deux suites de nombre réels :

$$- (\exists N \in \mathbb{N}) \text{ et } (\exists c > 0) : \forall n > N \quad |v_n| < c|u_n|$$

$$\text{on écrit } v_n = O(u_n)$$

$$- \text{si } (\forall \epsilon > 0) \exists N \in \mathbb{N} : \forall n > N \quad |v_n| < \epsilon|u_n|$$

$$\text{on écrit } v_n = o(u_n)$$

$$- \text{si } \lim_{n \rightarrow \infty} \frac{v_n}{u_n} = c \quad \begin{array}{l} \text{si } c = 0 \quad v_n = o(u_n) \\ \text{si } c \neq 0 \quad v_n = O(u_n) \end{array}$$

Supposons maintenant que  $\{u_n\}$  tende vers  $u$  et  $\{v_n\}$  vers  $v$ .

$$\text{Notons } \Delta u_n = u_{n+1} - u_n$$

Définition 1 : on dit que  $\{u_n\}$  est d'ordre  $r$  si

$$|u_{n+1} - u| = O(|u_n - u|^r) \text{ et si } |u_n - u|^r = O(|u_{n+1} - u|).$$

Théorème 1 : soient  $\{u_n\}$  et  $\{v_n\}$  deux suites qui convergent respectivement vers  $u$  et  $v$ .

Si  $(\exists N \in \mathbb{N})$  et  $(\exists(\alpha, \beta))$  vérifiant  $\alpha < 1 < \beta$  tels que  $\forall n \in \mathbb{N}$   
 $\frac{v_{n+1} - v}{v_n - v} \notin [\alpha, \beta]$  et si  $\lim_{n \rightarrow \infty} \frac{v_n - u}{v_n - v} = a$  alors  $\lim_{n \rightarrow \infty} \frac{\Delta u_n}{\Delta v_n} = a$ .

La réciproque n'est pas vraie:  $u_n = 1/n$  et  $v_n = (-1)^n/n$ .

Théorème 2 : soient  $\{u_n\}$  et  $\{v_n\}$  deux suites qui convergent respectivement vers  $u$  et  $v$ . Alors :

$$u_n - u = O(v_n - v) \iff \Delta u_n = O(\Delta v_n).$$

Théorème 3 : soient  $\{u_n\}$  et  $\{v_n\}$  deux suites qui convergent respectivement vers  $u$  et  $v$ . Alors :

$$u_n - u = o(v_n - v) \implies \Delta u_n = o(\Delta v_n).$$

Théorème 4 : soient  $\{u_n\}$  et  $\{v_n\}$  deux suites telles que :

$$u_{n+1} - u = O(u_n - u) \text{ et } v_n - v = o(u_n - u) \text{ alors}$$

$$v_n - v = o(u_{n-k} - u) \quad \forall k \geq 0.$$

Théorème 5 : soient  $u_n$  et  $v_n$  deux suites telles que

$$u_n - u = O(u_{n+1} - u) \text{ et } v_n - v = o(u_n - u) \text{ alors}$$

$$v_n - v = o(u_{n+k} - u) \quad \forall k \geq 0.$$

Théorème 6 : si  $u_{n+1} - u = O(u_n - u)$  et  $v_n - v = O(u_n - u)$  alors

$$v_n - v = O(u_{n-k} - u) \quad \forall k \geq 0.$$

Théorème 7 : si  $u_n - u = O(u_{n+1} - u)$  et  $v_n - v = O(u_n - u)$  alors

$$v_n - v = O(u_{n+k} - u) \quad \forall k \geq 0.$$

Théorème 8 : soient  $\{u_n\}$  et  $\{v_n\}$  deux suites qui convergent respectivement vers  $u$  et  $v$ . Si  $\{v_n - v\}$  est strictement monotone et si

$$\lim_{n \rightarrow \infty} \frac{\Delta v_n}{\Delta v_n} = a \text{ alors } \lim_{n \rightarrow \infty} \frac{u_n - u}{v_n - v} = a.$$

La propriété principale des algorithmes que nous allons étudier est d'accélérer la convergence d'une suite de scalaires. L'expression accélération de la convergence est à mettre ici entre guillemets car je ne suppose pas nécessairement que la suite initiale converge et cette expression signifie également "ralentissement de la divergence" ou extension du rayon de convergence comme le montre le résultat suivant :

Théorème 9 : soient  $\{S_n\}$  et  $\{V_n\}$  deux suites convergentes telles que

$$\lim_{n \rightarrow \infty} |\Delta S_n|^{1/n} = 1/R_1 \text{ et } \lim_{n \rightarrow \infty} |\Delta V_n|^{1/n} = 1/R_2 \text{ avec } R_1 < R_2. \text{ Alors } \{V_n\}$$

converge plus vite que  $\{S_n\}$  en ce sens que  $\lim_n \frac{\Delta v_n}{\Delta S_n} = 0$ .

## 1.2 DÉFINITION.

Il me faut définir maintenant le terme "accélération de la convergence"; je vais donner en fait deux définitions dont l'une ne nécessite pas la convergence. Le terme "accélération de la convergence" est équivalent ici à l'expression "converge plus vite".

Définition 1 : soient deux suites  $\{u_n\}$  et  $\{v_n\}$  qui convergent respectivement vers  $u$  et  $v$  et  $\lim_{n \rightarrow \infty} \frac{v_n - v}{u_n - u} = c$  si  $c = 0$  nous dirons que  $v_n$  converge plus vite que  $u_n$ .

Définition 3 : soient deux suites  $\{u_n\}$  et  $\{v_n\}$  telles que  $\lim_n \frac{\Delta v_n}{\Delta u_n} = c$  :  
 si  $c = 0$  nous dirons que  $v_n$  converge plus vite que  $u_n$ .

Il nous faut également définir ce qu'est un procédé régulier.

Définition 4 : soit  $T$  une méthode qui transforme une suite  $\{u_n\}$  convergente et une suite  $\{v_n\}$ . Nous dirons que  $T$  est régulière si  $\lim_{n \rightarrow \infty} u_n = \lim_{n \rightarrow \infty} v_n$  quelle que soit la suite  $u_n$  initiale.

Remarque : on peut trouver un exposé des résultats de ce chapitre dans les références [3,12,13].

## CHAPITRE II

ALGORITHME DE TYPE  $\theta$ 

Ce chapitre est consacré à l'étude des algorithmes de type  $\theta$  dans toute leur généralité. Après avoir énoncé quelques propriétés, nous étudions particulièrement la deuxième colonne du tableau  $\theta$  pour laquelle nous donnons des conditions de conservation de la limite et d'accélération de la convergence reposant sur le choix des suites auxiliaires  $\{b_n\}$ ,  $\{a_k^{(n)}\}$ . Nous faisons ensuite l'étude de la stabilité numérique et la propagation des erreurs. Tous ces résultats conduisent à la construction des suites  $\{b_n\}$  et  $\{a_k^{(n)}\}$ .

## II.1 FORMULATION GÉNÉRALE.

Dans cet exposé, les indices  $k$  et  $n$  prennent leurs valeurs sur l'ensemble  $N$  des entiers positifs ou nuls.

Brezinski a noté que divers algorithmes de transformation non linéaire de suites de nombres réels ont une définition commune.

Soit  $\{S_n\}$  une suite de nombres réels. On initialise :

$$\theta_{-1}^{(n)} = b_n \text{ et } \theta_0^{(n)} = S_n \quad \forall n = 0, 1, \dots$$

et on calcule les quantités  $\theta_k^{(n)}$  au moyen des relations :

$$(1) \quad \theta_{k+1}^{(n)} = \theta_{k-1}^{(n+1)} + a_k^{(n)} [\theta_k^{(n+1)} - \theta_k^{(n)}]^{-1} \quad \forall k, n = 0, 1, \dots$$

dans lesquelles le choix de  $b_n$  et  $a_k^{(n)}$  caractérise l'algorithme.

Nous dirons que cet algorithme est l'algorithme de type  $\theta$  associée aux nombres  $b_n$  et  $a_k^{(n)}$  et nous le noterons  $\theta(b_n, a_k^{(n)})$ -algorithme.

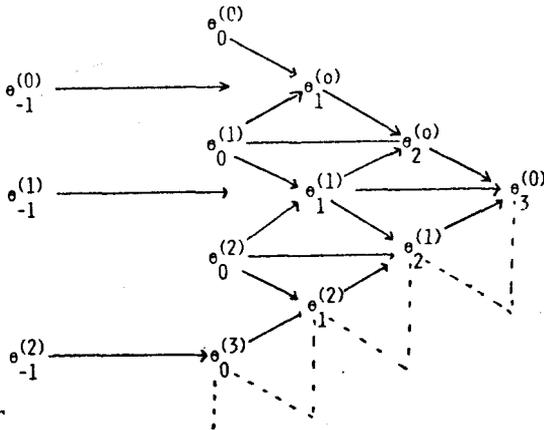
Pour alléger les notations, introduisons l'opérateur de différence définie par :

$$\Delta^0 S_n = S_n \text{ et } \Delta^{k+1} S_n = \Delta^k S_{n+1} - \Delta^k S_n.$$

Lorsque  $\Delta$  s'applique à des quantités dépendant de deux indices  $n$  et  $k$ , il agit toujours sur l'indice  $n$ .

Ces algorithmes sont des algorithmes récursifs dit de losange. On place les quantités dans un tableau à double entrée : tableau  $\theta$  ; l'indice inférieur  $k$  représente une colonne et l'indice supérieur  $n$  une diagonale descendante.

La relation (1) relie, dans ce tableau, des quantités situées aux quatre sommets d'un losange ; la quantité la plus à droite est calculée à partir des trois autres comme l'indiquent les flèches dans le schéma suivant :



Dans la suite nous ferons intervenir une sous-classe importante des algorithmes de type  $\theta$  ; celle où  $b_n = 0 \forall n$  et où  $a_k^{(n)} = \alpha_k$  indépendant de  $n$ . On ne restreint pas la généralité de la transformation en imposant  $\alpha_0 = 1$ . Cette sous classe est appelée algorithme de type  $\gamma$  associé à la suite  $\alpha_k$  (Cordellier [22]).

Les colonnes qui nous intéressent du point de vue accélération sont les colonnes paires, les colonnes impaires sont de simples intermédiaires de calcul. Nous étudierons donc les suites  $\{\theta_{2k}^{(n)}\}$  où  $\{\theta_k^{(n)}\}$  est la transformée de la suite initiale par les  $k$  premières étapes de l'algorithme ( $k$  fixé).

Nous dirons que :

- deux algorithmes de type  $\theta$  sont équivalents si les colonnes paires sont égales

- un procédé  $T$  de transformation de suites, fournissant les suites  $\{T_k^{(n)}\}$ , est équivalent à un algorithme type  $\theta$  si  $\theta_{2k}^{(n)} = T_k^{(n)}$  pour tout  $k$  et  $n$  positifs ou nuls.

Nous retrouvons ainsi des algorithmes connus : plaçons nous dans le cas  $b_n$  constant ; dans ce cas quelle que soit la constante les colonnes paires ne dépendent pas de cette constante, donc on ne restreint pas la généralité en prenant  $b_n = 0 \forall n$ . Alors :

1.  $a_k^{(n)} = 1 \forall k, n$  nous avons l' $\epsilon$ -algorithme de Wynn

2.  $a_k^{(n)} = \Delta_n \forall k, n$  la première généralisation de l' $\epsilon$ -algorithme (la suite  $\{x_n\}$  est une suite auxiliaire)

3.  $a_k^{(n)} = \Delta x_{n+k} \forall n, k$  la deuxième généralisation de l'ε-algorithme

4.  $a_k^{(n)} = x_{n+k+1} - x_n \forall k, n$  le ρ-algorithme

5.  $a_{2k}^{(n)} = 1$

$a_{2k}^{(n)} = \Delta \theta_{2k}^{(n+1)} \Delta \theta_{2k+1}^{(n)} \Delta \theta_{2k+1}^{(n+1)} / \Delta \theta_{2k+1}^{2(n)}$   $\forall n, k$ ; de θ-algorithme Brezinski [3]

6.  $a_{2k}^{(n)}$  quelconque

$a_{2k+1}^{(n)} = \Delta \theta_{2k}^{(n)} \Delta \theta_{2k+1}^{(n)} \frac{x_{n+k+1}}{x_n - x_{n+k+1}} \forall n, k$  } procédé d'extrapolation  
de Richardson [47]

7.  $a_{2k}^{(n)}$  quelconque

$a_{2k+1}^{(n)} = \Delta \theta_{2k}^{(n)} \Delta \theta_{2k+1}^{(n)} \frac{(\Delta S_{n+k+1})^{k+1}}{(\Delta S_{n+k})^{k+1} - (\Delta S_{n+k+1})^{k+1}} \forall n, k$  } Overholt [42]

et de manière générale tout algorithme qui a l'étape k se met sous la forme :

$$T_k^{(n)} = \frac{r_k^{(n)} T_{k-1}^{(n+1)} - r_k^{(n+1)} T_{k-1}^{(n)}}{r_k^{(n)} - r_k^{(n+1)}} \forall n, k \text{ avec } T_0^{(n)} = S_n$$

nous pouvons mettre cet algorithme sous la forme de type θ avec

$a_{2k}^{(n)}$  quelconque

$$a_{2k+1}^{(n)} = \Delta \theta_{2k}^{(n)} \Delta \theta_{2k+1}^{(n)} \frac{r_k^{(n+1)}}{r_k^{(n)} - r_k^{(n+1)}} \forall k, n$$

Remarque : 5 n'est pas un algorithme de losange car il faut quatre termes

$\theta_{2k}^{(n)}, \theta_{2k}^{(n+2)}, \theta_{2k}^{(n+3)}$  pour calculer  $\theta_{2k+2}^{(n)}$ .

## II.2 PROPRIÉTÉS.

Avant de donner le moindre résultat pour ces algorithmes il faut préciser un point important. On voit que lorsqu'on applique un algorithme de type  $\theta$  à une suite  $\{S_n\}$  il se peut que deux quantités voisines  $\theta_k^{(n)}$  et  $\theta_k^{(n+1)}$  deviennent égales. Il est alors impossible de continuer à construire le tableau  $\theta$  car il y aurait division par zéro. Dans la suite la convergence devra être comprise avec la restriction calquée sur celle énoncée par Wynn [65] : "bien que des conditions spéciales puissent être imposées à la suite  $\{S_n\}$  pour éviter cette division par zéro, dans l'exposition d'une théorie générale où l'on n'impose aucune condition sur la suite initiale, les résultats énoncés ne concernent que les nombres qui peuvent être calculés".

Ce type d'algorithme est une transformation non linéaire de suite à suite, c'est-à-dire que si on l'applique à la somme terme à terme de deux suites les quantités  $\theta_k^{(n)}$  que l'on obtient ne sont pas les sommes des quantités obtenues en appliquant le  $\theta(k_n, a_k^{(n)})$  algorithme séparément à chacune des suites. Cependant nous avons :

Propriété 1 : soit  $c$  une constante. Si l'application du  $\theta(c, a_k^{(n)})$ -algorithme à  $\{S_n\}$  et à  $\{aS_n + B\}$  fournit respectivement les quantités  $\theta_k^{(n)}$  et  $\bar{\theta}_k^{(n)}$  alors :

$$\bar{\theta}_{2k}^{(n)} = a\theta_{2k}^{(n)} + b \quad \forall k, n$$

$$\bar{\theta}_{2k+1}^{(n)} = \theta_{2k+1}^{(n)} / a$$

Démonstration : elle se fait par récurrence.

$$\left. \begin{aligned} \theta_1^{(n)} &= \frac{a_0^{(n)}}{\Delta S_n} \\ \bar{\theta}_1^{(n)} &= \frac{a_0^{(n)}}{a\Delta S_n} \end{aligned} \right\} \Rightarrow \bar{\theta}_1^{(n)} = \theta_1^{(n)} / a$$

$$\theta_2^{(n)} = \frac{a_1^{(n)}}{\Delta\theta^{(n)}} + S_{n+1}$$

$$\begin{aligned} \bar{\theta}_2^{(n)} &= \frac{a_1^{(n)}}{\Delta\bar{\theta}_1^{(n)}} + a S_{n+1} + b = a \left[ \frac{a_1^{(n)}}{\Delta\theta_1^{(n)}} + S_{n+1} \right] + b \\ &= a \theta_2^{(n)} + b \end{aligned}$$

Supposons qu'à l'étape j nous ayons :

$$\bar{\theta}_{2j}^{(n)} = a \theta_{2j}^{(n)} + b \quad \forall n = 0, 1, \dots$$

$$\bar{\theta}_{2j+1}^{(n)} = \theta_{2j+1}^{(n)} / a \quad \forall n = 0, 1, \dots$$

$$\text{alors } \bar{\theta}_{2j+2}^{(n)} = \bar{\theta}_{2j}^{(n+1)} + \frac{a_{2j+1}^{(n)}}{\Delta\bar{\theta}_{2j+1}^{(n)}}$$

$$\text{d'où } \bar{\theta}_{2j+2}^{(n)} = a\theta_{2j}^{(n+1)} + b + a \frac{a_{2j+1}^{(n)}}{\Delta\theta_{2j+1}^{(n)}} = a \left( \theta_{2j}^{(n+1)} + \frac{a_{2j+1}^{(n)}}{\Delta\theta_{2j+1}^{(n)}} \right) + b$$

$$\bar{\theta}_{2j+2}^{(n)} = a\theta_{2j+2}^{(n+1)} + b$$

$$\bar{\theta}_{2j+3}^{(n)} = \bar{\theta}_{2j+1}^{(n+1)} + \frac{a_{2j+2}^{(n)}}{\Delta\bar{\theta}_{2j+2}^{(n)}} = \frac{\theta_{2j+1}^{(n+1)}}{a} + \frac{a_{2j+2}^{(n)}}{a\Delta\theta_{2j+2}^{(n)}}$$

$$\bar{\theta}_{2j+3}^{(n)} = \bar{\theta}_{2j+1}^{(n+1)} + \frac{a_{2j+2}^{(n)}}{\theta_{2j+1}^{(n)}} = \frac{\theta_{2j+3}^{(n)}}{a}$$

Remarque : nous pouvons généraliser ce résultat au cas d'une suite auxiliaire  $b_n$  non constante : soit  $d$  une constante réelle. Si l'application du  $\theta(b_n, a_k^{(n)})$ -algorithme à  $\{S_n\}$  et du  $\theta(\frac{b}{a} + d, a_k^{(n)})$  à  $\{aS_n + b\}$  fournissent respectivement  $\theta_k^{(n)}$  et  $\bar{\theta}_k^{(n)}$  alors :

$$\bar{\theta}_{2k}^{(n)} = a\theta_{2k}^{(n)} + b$$

$$\bar{\theta}_{2k+1}^{(n)} = \frac{\theta_{2k+1}^{(n)}}{a} + d \quad \forall n, k = 1, 2, \dots$$

Propriété 2 :

$$a_{k+1}^{(n-1)} [\theta_{k+2}^{(n-1)} - \theta_k^{(n)}]^{-1} - a_{k-1}^{(n)} [\theta_k^{(n)} - \theta_{k-2}^{(n-1)}]^{-1} = a_k^{(n)} [\theta_k^{(n+1)} - \theta_k^{(n)}]^{-1} - a_k^{(n-1)} [\theta_k^{(n)} - \theta_k^{(n-1)}]^{-1}$$

Démonstration : on a les relations :

$$\theta_{k+1}^{(n-1)} - \theta_{k-1}^{(n)} = a_k^{(n-1)} [\theta_k^{(n)} - \theta_k^{(n-1)}]^{-1}$$

$$\theta_{k+1}^{(n)} - \theta_{k+1}^{(n+1)} = a_k^{(n)} [\theta_k^{(n+1)} - \theta_k^{(n)}]^{-1}$$

soustrayons et rangeons les termes à gauche du signe égal

$$(\theta_{k+1}^{(n)} - \theta_{k+1}^{(n-1)}) - (\theta_{k-1}^{(n+1)} - \theta_{k-1}^{(n)}) = a_k^{(n)} [\theta_k^{(n+1)} - \theta_k^{(n)}]^{-1} - a_k^{(n-1)} [\theta_k^{(n)} - \theta_k^{(n-1)}]^{-1}$$

nous avons aussi

$$\begin{aligned} \theta^{k+2}(n-1) - \theta^k(n) &= a^{k+1}(n-1) [\theta^{k+1}(n) - \theta^{k+1}(n-1)] \\ \theta^{k+1}(n-1) - \theta^{k-1}(n) &= a^{k-1}(n) [\theta^k(n) - \theta^k(n-1)] \end{aligned}$$

d'où la relation cherchée.

Comme on le voit cette propriété permet de passer de colonnes paires à colonnes paires dans le tableau  $\theta$ . En partant d'une relation semblable, Cordellier [2,33] a donné une interprétation géométrique de l'algorithme ainsi que des règles particulières de calcul de l'algorithme (lorsque deux ou plusieurs valeurs voisines deviennent égales).

Si nous appliquons un algorithme de type  $\theta$  à une suite de scalaires nous avons les relations :

$$\begin{aligned} \theta^{2k+2}(n) - \theta^{2k}(n) &= \frac{a^{2k+1}(n)}{a^{2k}(n+1)} + \frac{\Delta b^{n+1} + \frac{\Delta s^{n+1}}{a} - \frac{\Delta s^n}{a}}{a} \\ \theta^{2k+2}(n) - \theta^{2k}(n) &= \frac{a^{2k+1}(n)}{a^{2k}(n+1)} + \frac{\Delta b^{n+1} + \frac{\Delta s^{n+1}}{a} - \frac{\Delta s^n}{a}}{a} \end{aligned}$$

Nous appellerons algorithme de type  $\theta_2$  la restriction à la deuxième colonne d'un tableau  $\theta$ .

On peut ainsi remarquer que le passage de la colonne  $2k$  à la colonne  $2k+2$  apparait comme un cas particulier de l'algorithme de type  $\theta_2$  où :

$$\Delta b_{n+1} = \Delta \theta_{2k-1}^{(n+1)}, a_{2k}^{(n)} = a_0^{(n)} \quad \text{et} \quad a_{2k+1}^{(n)} = a_1^{(n)} \quad \forall n=0,1,\dots$$

Cette remarque est fondamentale pour la suite de cet exposé ; en effet il nous suffit d'étudier les algorithmes de type  $\theta_2$ , les théorèmes démontrés pour ces algorithmes donneront des théorèmes pour les algorithmes de type  $\theta$  ; de plus nous pouvons faire la même chose pour les applications répétées.

### II.3 ALGORITHME DE TYPE $\theta_2$

La remarque précédente montre l'intérêt qu'il y a à étudier les algorithmes de type  $\theta_2$ . Nous notons le  $\theta_2(b_n, a_k^{(n)})$ -algorithme, l'algorithme de type  $\theta_2$  associé aux suites de nombres  $b_n, a_0^{(n)}, a_1^{(n)}$ .

Dans l'étude de la convergence nous nous bornerons aux suites d'ordre 1 c'est-à-dire : les suites  $\{S_n\}$  telles qu'il existe  $K_1$  et  $K_2$  vérifiant :

$$0 < K_2 < \frac{|S_{n+1} - S|}{|S_n - S|} < K_1 < \infty \quad \forall n > N$$

### II.3.1 THEOREMES DE CONVERGENCE.

Nous allons d'abord caractériser le type de suites pour lesquelles les algorithmes de type  $\theta_2$  convergent en une étape.

Théorème 10 : une condition nécessaire et suffisante pour que

$\theta_2^{(n)}(c, a_k^{(n)}) = S \quad \forall n > N$  est que :

$$S_n = S + A \prod_{i=0}^{n-1} (1 + \rho_i) \quad \forall n > N$$

avec

$$\rho_0 = \frac{a_0^{(0)}}{B}$$

$$\rho_i = \frac{a_0^{(i)}}{B + \sum_{j=0}^{i-1} (a_0^{(j)} - a_1^{(j)})} \quad i = 1, 2, \dots$$

$$A = S_0 - S$$

Démonstration : D'après la propriété 1 nous pouvons dire que si le résultat est vrai pour  $\{S_n - S\}$  il sera vrai pour  $\{S_n\}$ .

Appliquons donc le  $\theta_2(c, a_k^{(n)})$ -algorithme à la suite  $x_n = S_n - S$

$$\theta_2^{(n)} = x_{n+1} + \frac{a_1^{(n)}}{\Delta \theta_1^{(n)}} = x_{n+1} + \frac{a_1^{(n)}}{\frac{a_0^{(n+1)}}{\Delta x_{n+1}} - \frac{a_0^{(n)}}{\Delta x_n}}$$

$$\theta_2^{(n)} = x_{n+1} + \frac{a_1^{(n)} (\Delta x_{n+1} \Delta x_n)}{a_0^{(n+1)} \Delta x_n - a_0^{(n)} \Delta x_{n+1}}$$

$$\theta_2^{(n)} = \frac{a_0^{(n+1)} x_{n+1} \Delta x_n + a_0^{(n)} x_{n+1} \Delta x_{n+1} + a_1^{(n)} \Delta x_{n+1} \Delta x_n}{a_0^{(n+1)} \Delta x_n - a_0^{(n)} \Delta x_{n+1}}$$

$$\theta_2^{(n)} = \frac{a_0^{(n+1)} x_{n+1} \Delta x_n - a_0^{(n)} x_{n+1} \Delta x_{n+1} + a_1^{(n)} x_{n+1} \Delta x_{n+1} - a_1^{(n)} \Delta x_n \Delta x_{n+1}}{a_0^{(n+1)} \Delta x_n - a_0^{(n)} \Delta x_{n+1}}$$

j'ajoute et je retranche au numérateur  $a_0^{(n)} x_n \Delta x_{n+1}$  et je regroupe les termes

$$\theta_2^{(n)} = \frac{a_0^{(n+1)} x_{n+1} \Delta x_n + (a_1^{(n)} - a_0^{(n)}) x_{n+1} \Delta x_{n+1} - a_0^{(n)} x_n \Delta x_{n+1} + x_n \Delta x_{n+1} (a_0^{(n)} - a_1^{(n)})}{a_0^{(n+1)} \Delta x_n - a_0^{(n)} \Delta x_{n+1}}$$

$$\theta_2^{(n)} = \frac{a_0^{(n+1)} x_{n+1} \Delta x_n - a_0^{(n)} x_n \Delta x_{n+1} + (a_1^{(n)} - a_0^{(n)}) \Delta x_n \Delta x_{n+1}}{a_0^{(n+1)} \Delta x_n - a_0^{(n)} \Delta x_{n+1}}$$

Je divise le numérateur et le dénominateur par  $\Delta x_n \Delta x_{n+1}$ .

$$\theta_2^{(n)} = \frac{a_0^{(n+1)} \frac{x_{n+1}}{\Delta x_{n+1}} - a_0^{(n)} \frac{x_n}{\Delta x_n} + a_1^{(n)} - a_0^{(n)}}{\frac{a_0^{(n+1)}}{\Delta x_{n+1}} - \frac{a_0^{(n)}}{\Delta x_n}}$$

$$\theta_2^{(n)} = \frac{\Delta \left[ \begin{array}{c} a_0^{(n)} \\ \frac{x_n}{\Delta x_n} \end{array} \right] + a_1^{(n)} - a_0^{(n)}}{\Delta \left[ \begin{array}{c} a_0^{(n)} \\ \frac{x_n}{\Delta x_n} \end{array} \right]}$$



posons 
$$\rho_n = a_0^{(n)} \frac{x_n}{\Delta x_n}$$

Nous pouvons dire que  $\theta_2(c, a_k^{(n)})$  converge en une itération  $\forall n > N$  si, appliqué à la suite  $\{S_n - S\}$ ,  $\theta_2^{(n)} = 0$  quel que soit  $n > N$ , ceci est vrai si et seulement si

$$\Delta \rho_n + (a_1^{(n)} - a_0^{(n)}) = 0 \quad \forall n > N$$

et

$$\Delta \left( \frac{a_0^{(n)}}{\Delta x_n} \right) \neq 0$$

or le fait que  $\Delta \left( \frac{a_0^{(n)}}{\Delta x_n} \right) = 0$  nous interdit d'appliquer le  $\theta_2(c, a_0^{(n)})$ .  
Nous nous restreignons donc à :

$$\Delta \rho_n + (a_1^{(n)} - a_0^{(n)}) = 0$$

notons  $a_1^{(n)} - a_0^{(n)} = c_n$

$$\Rightarrow \rho_{n+1} = \rho_n + c_n$$

$$\rho_{n+1} = \rho_0 + \sum_{i=0}^n c_i$$

$$\text{or } \frac{a_0^{(n)} x_n}{\Delta x_n} = \rho_n \Rightarrow x_{n+1} = x_n \left( \frac{a_0^{(n)} + \rho_n}{\rho_n} \right)$$

$$x_{n+1} = x_0 \prod_{i=0}^n \left( 1 + \frac{a_0^{(i)}}{\rho_i} \right)$$

$$\text{avec } \rho_n = \rho_0 + \prod_{i=0}^{n-1} (a_1^{(i)} - a_0^{(i)}) \quad n \geq 1$$

donc pour la suite  $\{S_n\}$  qui converge vers S

$$S_{n+1} = S + A \prod_{i=0}^n \left( 1 + \frac{a_0^{(i)}}{\rho_i} \right)$$

$$A = S_0 - S$$

Remarque : 1) nous donnons dans cette démonstration une autre manière de noter le  $\theta_2$  algorithme en fonction de la suite  $\{S_n\}$

$$\theta_2^{(n)} = \frac{\Delta(a_0^{(n)} \frac{x_n}{\Delta x_n}) + (a_1^{(n)} - a_0^{(n)})}{\Delta(\frac{a_0^{(n)}}{\Delta x_n})}$$

dans le cas où  $a_1^{(n)} = a_0^{(n)} = 1$  nous retrouvons la notation de Cordellier [22] pour l' $\varepsilon$ -algorithme.

2) les suites obtenues au théorème 10 ne sont pas nécessairement convergentes; cependant le  $\theta_2$  algorithme donne  $S$ . Dans le cas d'une suite divergente  $S$  sera appelé "anti limite".

Cependant les algorithmes de type  $\theta_2$  présentent un grand intérêt pour bien d'autres suites que celles du théorème 10. Et nous allons maintenant donner des conditions, sur les suites auxiliaires, qui permettent d'obtenir une bonne approximation de la limite d'une suite.

Il nous faut d'abord donner des conditions de conservation de la limite.

Théorème 11 : si nous appliquons le  $\theta_2(a_n, a_k^{(n)})$  à la suite  $\{S_n\}$  de limite  $S$ . Une condition nécessaire et suffisante pour que

$$\lim_{n \rightarrow \infty} S_n = \lim_{n \rightarrow \infty} \theta_2^{(n)} \text{ est que : } \lim_{n \rightarrow \infty} \frac{a_1^{(n)}}{\Delta \theta_1^{(n)}} = 0.$$

Ce résultat n'est pas très pratique ; en effet on peut construire  $a_1^{(n)} = U_n \Delta \theta_1^{(n)}$  avec  $\{U_n\}$  convergeant vers zéro ; mais il est difficile, dans ce cas, pour  $U_n$  de vérifier les conditions d'accélération (il faut connaître  $S$ ).

Ainsi, dans la suite de cette exposé, nous nous placerons dans les hypothèses suivantes :

$$H0 : a_0^{(n)} \neq 0, \frac{a_0^{(n+1)}}{a_0^{(n)}} = b_0^{(n)} \quad \lim_{n \rightarrow \infty} b_0^{(n)} = b_0$$

$$\lim_{n \rightarrow \infty} \frac{S_{n+1} - S_n}{S_n - S} = K_0 \neq 0$$

$$\frac{\Delta b_{n+1} \Delta S_{n+1}}{a_0^{(n)}} = A_0^{(n)} \quad \lim_{n \rightarrow \infty} A_0^{(n)} = A_0$$

$$H1 : \frac{a_1^{(n)}}{a_0^{(n)}} = c_0^{(n)} \quad \lim_{n \rightarrow \infty} c_0^{(n)} = c_0 \neq 0$$

Dans ces conditions nous pouvons donner une condition suffisante de régularité :

Théorème 12 : si nous appliquons le  $\theta_2(b_n, a_k^{(n)})$ -algorithme à la suite de nombres réels  $\{S_n\}$  de limite  $S$ . Si  $\left\{ \frac{a_1^{(n)}}{a_0^{(n)}} \right\}$  est bornée alors :

$$A_0 + b_0 - K_0 \neq 0 \implies \lim_{n \rightarrow \infty} S_n = \lim_{n \rightarrow \infty} \theta_2^{(n)} = S$$

Démonstration : écrivons la relation (1)

$$\theta_2^{(n)} = S_{n+1} + \frac{a_1^{(n)}}{\Delta \theta_1} = S_{n+1} + \frac{a_1^{(n)}}{\Delta b_{n+1} + \frac{a_0^{(n+1)}}{\Delta S_{n+1}} - \frac{a_0^{(n)}}{\Delta S_n}}$$

$$a_0^{(n)} \neq 0 \quad \forall n$$

$$\theta_2^{(n)} = S_{n+1} + \frac{\frac{a_1^{(n)}}{a_0^{(n)}} \Delta S_{n+1}}{\frac{\Delta b_{n+1} \Delta S_{n+1}}{a_0^{(n)}} + \frac{a_0^{(n+1)}}{\Delta S_{n+1}} - \frac{\Delta S_{n+1}}{\Delta S_n}} \quad \text{d'où le résultat ...}$$

La seule condition contraignante est que  $\frac{S_{n+1}-S}{S_n-S}$  possède une limite et nous pouvons toujours choisir  $\{b_n\}$ ,  $\{a_o^{(n)}\}$  vérifiant les conditions du théorème.

Si maintenant nous nous plaçons dans l'hypothèse H1 nous pouvons énoncer le résultat :

Théorème 13 : si nous appliquons le  $\theta_2(b_n, a_k^{(n)})$  algorithme à la suite de nombre réels  $\{S_n\}$  ayant pour limite S. Si  $-1 \leq K_o < 1$  et si  $K_o - b_o - A_o \neq 0$ . Alors

$$c_o = \frac{K_o - b_o - A_o}{K_o - 1} \iff \lim_{n \rightarrow \infty} \frac{\theta_2^{(n)} - S}{S_{n+1} - S} = 0$$

Démonstration :

$$\frac{\theta_2^{(n)} - S}{S_{n+1} - S} = \frac{S_{n+1} - S}{S_{n+1} - S} + \frac{\frac{a_1^{(n)}}{\Delta \theta_1^{(n)}} (S_{n+1} - S)}{\Delta \theta_1^{(n)}}$$

$$\frac{\theta_2^{(n)} - S}{S_{n+1} - S} = 1 + \frac{\frac{a_1^{(n)}}{(S_{n+1} - S)}}{\Delta b_{n+1} + \frac{a_o^{(n+1)}}{\Delta S_{n+1}} - \frac{a_o^{(n)}}{\Delta S_n}}$$

$$\frac{\theta_2^{(n)} - S}{S_{n+1} - S} = 1 + \frac{\frac{a_1^{(n)}}{S_{n+1} - S} \times \frac{\Delta S_{n+1}}{a_o^{(n)}}}{\frac{\Delta b_{n+1} \Delta S_{n+1}}{a_o^{(n)}} + \frac{a_o^{(n+1)}}{a_o^{(n)}} - \frac{\Delta S_{n+1}}{\Delta S_n}}$$

$$\frac{\theta_2^{(n)} - S}{S_{n+1} - S} = 1 + \frac{\frac{a_1^{(n)} S_{n+2} - S}{(S_{n+2} - S) - 1}}{\frac{a_0^{(n)} S_{n+1} - S}{\Delta b_{n+1} \frac{\Delta S_{n+1}}{a_0^{(n)}} + \frac{a_0^{(n+1)}}{a_0^{(n)}} - \frac{\Delta S_{n+1}}{\Delta S_n}}}}$$

$$K_0 \neq 1 \Rightarrow \lim_{n \rightarrow \infty} \frac{\Delta S_{n+1}}{\Delta S_n} = K_0$$

d'où la limite :

$$\frac{c_0(K_0 - 1)}{A_0 + b_0 - K_0} = -1 \Rightarrow c_0 = \frac{K_0 - b_0 - A_0}{K_0 - 1} \text{ pour } K_0 \neq 1$$

Remarque : dans le théorème précédent nous avons étudié le rapport  $\frac{\theta_2^{(n)} - S}{S_{n+1} - S}$

ou pour le calcul de  $\theta_2^{(n)}$  nous utilisons  $S_n, S_{n+1}, S_{n+2}$  ; il faudrait donc

étudier  $\frac{\theta_2^{(n)} - S}{S_{n+2} - S}$  : je me suis borné dans cet exposé aux suites d'ordre 1,

dans ce cas

$$\lim_{n \rightarrow \infty} \frac{\theta_2^{(n)} - S}{S_{n+1} - S} = 0 \Rightarrow \lim_{n \rightarrow \infty} \frac{\theta_2^{(n)} - S}{S_{n+2} - S} = 0$$

Il n'est pas nécessaire que la suite soit convergente pour parler d'accélération de la convergence dans le cas où la suite diverge nous

appellerons  $K_0 = \lim_{n \rightarrow \infty} \frac{\Delta S_{n+1}}{\Delta S_n}$

Théorème 14 : si nous appliquons le  $\theta_2(b_n, a_k^{(n)})$ -algorithme à une suite  $\{S_n\}$ , si  $K_0 \neq 1$ ,  $a_0^{(n)} \neq 0 \forall n$  et si  $K_0 - b_0 - A_0 \neq 0$  alors :

$$c_0 = \frac{K_0 - b_0 - A_0}{K_0 - 1} \implies \lim_n \frac{\Delta\theta_2^{(n)}}{\Delta S_{n+1}} = 0$$

Démonstration :

$$\frac{\Delta\theta_2^{(n)}}{\Delta S_{n+1}} = \frac{\Delta S_{n+1} + \frac{a_1^{(n+1)}}{\Delta\theta_1^{(n+1)}} - \frac{a_1^{(n)}}{\Delta\theta_1^{(n)}}}{\Delta S_{n+1}}$$

$$\frac{\Delta\theta_2^{(n)}}{\Delta S_{n+1}} = 1 + \frac{\frac{a_1^{(n+1)}}{\Delta S_{n+1}} / \Delta S_{n+1}}{\Delta b_{n+2} + \frac{a_0^{(n+2)}}{\Delta S_{n+2}} - \frac{a_0^{(n+1)}}{\Delta S_{n+1}}} - \frac{\frac{a_1^{(n)}}{\Delta S_{n+1}} / \Delta S_{n+1}}{\Delta b_{n+1} + \frac{a_0^{(n+1)}}{\Delta S_{n+1}} - \frac{a_0^{(n)}}{\Delta S_n}}$$

puisque  $a_0^{(n)} \neq 0$

$$\frac{\Delta\theta_2^{(n)}}{\Delta S_{n+1}} = 1 + \frac{\frac{\frac{a_1^{(n+1)}}{a_0^{(n+1)}} \frac{\Delta S_{n+2}}{\Delta S_{n+1}}}{\frac{\Delta b_{n+2} \Delta S_{n+2}}{a_0^{(n+1)}} + \frac{a_0^{(n+2)}}{a_0^{(n+1)}} - \frac{\Delta S_{n+2}}{\Delta S_{n+1}}} - \frac{\frac{a_1^{(n)}}{a_0^{(n)}}}{\frac{\Delta b_{n+1} \Delta S_{n+1}}{a_0^{(n)}} + \frac{a_0^{(n+1)}}{a_0^{(n)}} - \frac{\Delta S_{n+1}}{\Delta S_n}}}{\Delta S_{n+1}}$$

d'où à la limite

$$1 + \frac{c_0 (K_0 - 1)}{A_0 + b_0 - K_0} = 0 \implies c_0 = \frac{-A_0 - b_0 + K_0}{K_0 - 1}$$

La remarque précédente est encore valable si  $K_0 \neq 0,1$  nous avons

$$\lim_{n \rightarrow \infty} \frac{\Delta\theta_2^{(n)}}{\Delta S_{n+2}} = 0.$$

Exemple n°1 : soit la suite  $S_n = \sum_{i=1}^n (-1)^i i^{-1/2}$ . Nous avons pour cette

suite  $K_0 \neq 1$   $K_0 = -1$ .

Nous pouvons prendre  $b_0^{(n)} = c_0^{(n)} = 1$

$S = 0,60489864216304$

$n$	$S_n$	$\frac{(n)}{1 \times 2}$
4	0,40	0,604
9	0,75	0,605
14	0,48	0,6048
19	0,71	0,6049
24	0,50	0,60487
29	0,69	0,60491

Et si nous itérons cet algorithme

$\frac{\varepsilon(n)}{2 \ 2}$	$\frac{\varepsilon(n)}{3 \ 2}$	$\frac{\varepsilon(n)}{4 \ 2}$
0,60485	0,60489867	0,604898645
0,60490	0,604898641	0,604898643
0,6048987	0,6048986436	0,6048986423
0,6048986	0,6048986433	0,60489864214
0,60489865	0,6048986434	

Exemple n°2 : soit  $S_n = n/(n+1)$  dans ce cas  $K_0 = 1$ . Bien que  $c_0^{(n)} = b_0^{(n)} = 1$  vérifiant la condition d'accélération nous constatons que si  $K_0 = 1$  nous ne pouvons rien dire :

$n$	$S_n$	$\theta_2^{(n)}$
4	0,83	0,90
9	0,90	0,95
14	0,93	0,96
19	0,95	0,97
24	0,96	0,98
29	0,96	0,98

Exemple n°3 : enfin voici un exemple d'une suite divergente qui par le  $\theta$ -algorithme donne une limite : c'est le cas de l'accroissement du rayon de convergence d'une série. Soit  $S_n = \sum_{i=1}^n \frac{(-2)^i}{i+1}$  qui est le développement en série de  $\log(3)$  hors du disque de convergence. Nous avons  $K_0 = -2$ .

Nous appliquons à  $S_n$  le  $\theta$ -algorithme avec  $c_0^{(n)} = 1 = b_0^{(n)}$  et nous itérons le procédé  $\log(3) = 0,0.9861228866$

$n$	$S_n$	$10^{(n)}$	$1^{\epsilon_2^{(n)}}$	$2^{\epsilon_2^{(n)}}$	$3^{\epsilon_2^{(n)}}$	
4	-0,7	$10^1$	1,06	1,097	1,0986	
9	+0,12	$10^3$	1,2	1,099	1,0985	
14	-0,26	$10^4$	-0,22	$10^1$	1,094	1,0987
19	0,65	$10^5$	0,19	$10^2$	1,12	1,097
24	-0,16	$10^7$	-0,30	$10^3$	-1,17	
29	0,98	$10^8$	0,50	$10^4$		

nous pouvons remarquer, : bien que chaque colonne diverge nous approchons de la limite. Si nous continuons, nous obtenons  $14^{\epsilon_2^{29}} = 1,09861228866$ .

## II.3.2 SUITES A CONVERGENCE LOGARITHMIQUE.

Dans l'étude précédente nous avons écarté le cas  $K_0=1$ , c'est-à-dire les suites à convergence logarithmique. On peut cependant se demander s'il est possible d'accélérer de telles suites à l'aide des algorithmes de type  $\theta$ .

Soit une suite de nombres réels  $\{S_n\}$  convergent vers  $S$ ; si cette suite est à convergence logarithmique nous pouvons dire que  $(S_{n+1} - S) / (S_n - S) = 1 + e_n$  avec  $\lim_{n \rightarrow \infty} e_n = 0$ . Nous pouvons remarquer que dans cette hypothèse nous n'avons pas nécessairement  $\lim_{n \rightarrow \infty} S_{n+1} / S_n = 1$  en effet :

$$\frac{\Delta S_{n+1}}{\Delta S_n} = \frac{S_{n+2} - S - (S_{n+1} - S)}{S_{n+1} - S - (S_n - S)}$$

$$\frac{\Delta S_{n+1}}{\Delta S_n} = \frac{\frac{S_{n+2} - S}{S_{n+1} - S} - 1}{1 - \frac{S_n - S}{S_{n+1} - S}} = \frac{e_{n+1}}{1 - \frac{1}{1+e_n}} = \frac{e_{n+1}}{e_n} (1 + e_n)$$

$$\Rightarrow \lim_{n \rightarrow \infty} \frac{S_{n+1}}{S_{n+1} - S} = \lim_{n \rightarrow \infty} \frac{e_{n+1}}{e_n} \quad (\text{si cette limite existe}).$$

Nous nous placerons pour cette étude au cas où  $\Delta b_n = 0$  :

Théorème 15 : si nous appliquons le  $\theta_2(c, a_k^{(n)})$ -algorithme à une suite  $(S_n)$  de limite  $S$ , avec  $\lim_n \frac{S_{n+1} - S}{S_n - S} = 1$

$$\left. \begin{aligned} b_0 &= \lim_{n \rightarrow \infty} \frac{e_{n+1}}{e_n} \\ c_0 &= \lim_{n \rightarrow \infty} \left( 1 + \frac{1}{e_n} - \frac{b_0^{(n)}}{e_{n+1}} \right) \end{aligned} \right\} \Leftrightarrow \begin{aligned} \lim_{n \rightarrow \infty} \theta_2^{(n)} &= S \\ \text{et } (\theta_2^{(n)} - S) &= o(S_{n+1} - S) \end{aligned}$$

Démonstration : si  $S_n$  est à convergence logarithmique alors  $\Delta S_{n+1} / (S_{n+1} - S)$  tend vers zéro quand  $n$  tend vers l'infini.

$$\begin{aligned} \frac{\theta_2^{(n)} - S}{S_{n+1} - S} &= 1 + \frac{a_1^{(n)} / (S_{n+1} - S)}{\frac{a_0^{(n+1)}}{\Delta S_{n+1}} - \frac{a_0^{(n)}}{\Delta S_n}} \\ &= 1 + \frac{\frac{a_1^{(n)}}{a_0^{(n)}} \frac{\Delta S_{n+1}}{S_{n+1} - S}}{\frac{a_0^{(n+1)}}{a_0^{(n)}} - \frac{\Delta S_{n+1}}{\Delta S_n}} \\ &= 1 + \frac{c_0^{(n)} e_{n+1}}{b_0^{(n)} - (1 + e_n) \frac{e_{n+1}}{e_n}} \end{aligned}$$

pour que  $c_0$  soit fini différent de zéro il faut et il suffit que :

$$b_0 = \lim_{n \rightarrow \infty} \frac{e_{n+1}}{e_n}$$

dans ce cas pour que  $(\theta_2^{(n)} - S) = o(S_{n+1} - S)$  il faut et il suffit que :

$$c_o = \lim_{n \rightarrow \infty} \left( 1 + \frac{1}{e_n} - \frac{b_o^{(n)}}{e_{n+1}} \right) \text{ lorsque cette limite existe.}$$

Corollaire : soit  $\{S_n\}$  une suite de nombres réels telle que

$\lim_{n \rightarrow \infty} S_n = S$  et  $\lim_{n \rightarrow \infty} \frac{S_{n+1} - S}{S_n - S} = 1$ . Il existe une famille de  $\theta_2$  algorithmes

tels que  $\theta_2^{(n)} = S \forall n > N$  ; il suffit, pour cela de prendre

$$c_o^{(n)} = 1 + \frac{1}{e_n} - \frac{b_o^{(n)}}{e_{n+1}}.$$

Démonstration : nous avons

$$\theta_2^{(n)} = S_{n+1} + \frac{\left(1 + \frac{1}{e_n} - \frac{b_o^{(n)}}{e_{n+1}}\right) \Delta S_{n+1}}{b_o^{(n)} - (1 + e_n) \frac{e_{n+1}}{e_n}}$$

$$\theta_2^{(n)} = S_{n+1} - \Delta S_{n+1} / e_{n+1} = S_{n+1} + S - S_{n+1}$$

$$\theta_2^{(n)} = S$$

C.Q.F.D.

Exemple :

Posons  $d_n = S_n - S$ . Soit une suite logarithmique telle que

$$d_{n+1} = d_n + a_1 d_n^p + a_2 d_n^{p+1} + \dots$$

$$\frac{d_{n+1}}{d_n} = 1 + a_1 d_n^{p-1} + a_2 d_n^p + \dots$$

dans ce cas  $e_n = a_1 d_n^{p-1} + a_2 d_n^p + \dots$

$$b_o = \lim_{n \rightarrow \infty} \frac{e_{n+1}}{e_n}$$

$$\frac{e_{n+1}}{e_n} \sim \frac{a_1 d_{n+1}^{p-1}}{a_1 d_n^{p-1}} \frac{[d_n + a_1 d_n^p]^{p+1}}{d_n^{p-1}}$$

$$\lim_{n \rightarrow \infty} \frac{e_{n+1}}{e_n} = +1 + a_n \quad \text{avec} \quad \lim_{n \rightarrow \infty} a_n = 0$$

prenons donc  $b_0^{(n)} = 1$

$$c_0^{(n)} = 1 - \left( \frac{1}{e_{n+1}} - \frac{1}{e_n} \right)$$

$$c_0^{(n)} \sim 1 - \left( \frac{1}{a_1 d_{n+1}^{p-1}} - \frac{1}{a_1 d_n^{p-1}} \right)$$

$$c_0^{(n)} \sim 1 - \frac{a_1 d_n^{p-1} - a_1 [d_n + a_1 d_n^p]^{p-1}}{a_1 [d_n + a_1 d_n^p]^{p-1} a_1 d_n^{p-1}}$$

$$\sim 1 + \frac{a_1^{2(p-1)} d_n^{2p-2}}{a_1^2 d_n^{2p-2}}$$

$$\Rightarrow \lim_{n \rightarrow \infty} c_0^{(n)} = p$$

pour ce genre de suites nous trouvons :

$$\theta_2^{(n)} = S_{n+1} - p \frac{\Delta S_n \Delta S_{n+1}}{\Delta^2 S_n}$$

On voit que dans ce cas si  $p=2$  on retrouve le  $\rho$ -algorithme simplifié.

Nous trouvons également une condition d'accélération de la convergence.

Corollaire : pour que le  $\theta_2(c, a_k^{(n)})$ -algorithme accélère la convergence il faut et il suffit que :

$$b_0 = \lim_{n \rightarrow \infty} \frac{e_{n+1}}{e_n}$$

et ) 
$$c_0 = \lim_{n \rightarrow \infty} \left( 1 + \frac{1}{e_n} - \frac{b_0^{(n)}}{e_{n+1}} \right)$$

Dans le cas  $b_0^{(n)} = 1 \forall n$  (algorithme de type  $\gamma_2$ ) nous obtenons comme condition :

$$c_0 = 1 - \lim_{n \rightarrow \infty} \Delta \left( \frac{1}{e_n} \right)$$

cette condition a déjà été démontré par Cordellier [19]. Nous pouvons également remarquer que dans le cas  $K_0 \neq 1$  la condition s'écrit  $b_0 = 1$   $c_0 = 1$  ce qui est le cas pour l' $\epsilon$ -algorithme.

Exemple 1 : soit la suite  $\{S_n\}$  obtenu par :  $S_{n+1} = S_n - 0,05 S_n^2$  ou  $S_1 = 0,938$ , nous avons  $K_0 = 1$  ; le calcul de  $c_0$  nous donne 2. Nous appliquons à  $S_n$  le  $\theta_2$ -algorithme  $b_0^{(n)} = b_0 = 1$  et  $c_0^{(n)} = c_0 = 2$ .  $S=0$

$n$	$S_n$	$\rho_2^{(n)}$	
5	0,91	- 0,21	$10^{-2}$
10	0,89	- 0,20	$10^{-2}$
15	0,87	- 0,19	$10^{-2}$
20	0,85	- 0,18	$10^{-2}$
25	0,84	- 0,17	$10^{-2}$

Dans la pratique on rencontre de nombreuses suites à convergence lente mais nous ne connaissons pas la limite  $S$  ; il nous est donc difficile d'utiliser ce qui précède aussi nous prendrons une autre définition de la convergence logarithmique des suites. Nous dirons que  $\{S_n\}$  est à convergence logarithmique si  $\lim_{n \rightarrow \infty} (\Delta S_{n+1} / \Delta S_n) = 1$  que nous pouvons encore écrire :

$$\frac{\Delta S_{n+1}}{\Delta S_n} = 1 + e_n \text{ avec } \lim_{n \rightarrow \infty} e_n = 0.$$

Si une suite  $S_n$  est à convergence lente et si de plus  $\lim_{n \rightarrow \infty} \Delta^2 S_n / \Delta S_n = 0$  alors la suite est à convergence logarithmique.

Pour l'étude de ce cas nous nous restreindrons au cas :

$$b_0^{(n)} = b_0, A_0^{(n)} = A_0 \text{ et } c_0^{(n)} = c_0 \quad \forall n$$

nous pouvons alors énoncer le résultat suivant :

Théorème 16 : soit  $\{S_n\}$  une suite à convergence logarithmique de limite  $S$ . Si nous lui appliquons le  $\theta_2(b_n, a_k^{(n)})$  algorithme.

Alors :

$$\left. \begin{array}{l} A_0 + b_0 - 1 = 0 \\ c_0 = \lim_{n \rightarrow \infty} \frac{1}{1 + \Delta\left(\frac{1}{e_n}\right)} \end{array} \right\} \Leftrightarrow \Delta\theta_2^{(n)} = o(\Delta S_{n+1})$$

de plus :

$$\lim_{n \rightarrow \infty} \frac{\Delta S_n \cdot \Delta S_{n+1}}{\Delta^2 S_n} = 0 \iff \lim_{n \rightarrow \infty} \theta_2^{(n)} = s$$

Démonstration :

$$\frac{\Delta \theta_2^{(n)}}{\Delta S_{n+1}} = \frac{\Delta S_{n+1} + \frac{a_1^{(n+1)}}{a_o^{(n+2)} + \frac{a_o^{(n+1)}}{\Delta S_{n+2}} - \frac{a_o^{(n+1)}}{\Delta S_{n+1}}} - \frac{a_1^{(n)}}{a_o^{(n+1)} + \frac{a_o^{(n)}}{\Delta S_{n+1}} - \frac{a_o^{(n)}}{\Delta S_n}}}{\Delta S_{n+1}}$$

$$\frac{\Delta \theta_2^{(n)}}{\Delta S_{n+1}} = 1 + \frac{\frac{a_1^{(n+1)}}{a_o^{(n+1)}} \frac{\Delta S_{n+2}}{\Delta S_{n+1}}}{\frac{\Delta b_{n+2}}{a_o^{(n+1)}} \frac{\Delta S_{n+2}}{\Delta S_{n+1}} + \frac{a_o^{(n+2)}}{a_o^{(n+1)}} - \frac{\Delta S_{n+2}}{\Delta S_{n+1}}} - \frac{\frac{a_1^{(n)}}{a_o^{(n)}}}{\frac{\Delta b_{n+1}}{a_o^{(n)}} \frac{\Delta S_{n+1}}{\Delta S_{n+1}} + \frac{a_o^{(n+1)}}{a_o^{(n)}} - \frac{\Delta S_{n+1}}{\Delta S_n}}$$

Pour que  $c_o$  soit différent de zéro il faut et il suffit que  $A_o + b_o - 1 = 0$

maintenant si  $A_o^{(n)} = A_o$ ,  $b_o^{(n)} = b_o$  et  $c_o^{(n)} = c_o$  alors pour que

$$\frac{\Delta \theta_2^{(n)}}{\Delta S_{n+1}} \xrightarrow[n \rightarrow \infty]{} 0 \text{ il faut et il suffit que}$$

$$c_o = \lim_{n \rightarrow \infty} \frac{1}{1 + \Delta\left(\frac{1}{e_n}\right)}$$

Pour accélérer ce type de suites il nous faut donc évaluer  $c_0$ .

Nous pouvons remarquer que si  $A_0 = 0$  et  $\lim_{n \rightarrow \infty} \frac{\Delta S_{n+1}}{\Delta S_n} \neq 1$  nous obtenons les conditions  $b_0 = 1$  et  $c_0 = 1$  qui est le  $\Delta^2$  d'Aitken.

Exemple 2 : soit la suite  $S_n = n/n+1$  nous prendrons

$$A_0 = 0 \Rightarrow b_0 = 1 \text{ et } b_0^{(n)} = 1$$

$$c_0 = \lim_{n \rightarrow \infty} \frac{1}{1 + \Delta\left(\frac{1}{e}\right)_n} = \frac{1}{1 + \frac{1}{\frac{n+1}{n+3} - 1} - \frac{1}{\frac{n}{n+2} - 1}}$$

$$c_0 = 2$$

ce qui nous donne le  $\rho_2$ -algorithme

n	$S_n$	$\rho_2^{(n)}$
5	0,857	0,9999999999999999 16 chiffres exacts
10	0,916	"
15	0,941	"
20	0,95	"
25	0,962	"

Exemple 3 : Il est possible de calculer  $c_0$  dans certains cas. Par

exemple pour  $\Delta S_n = 1/n^a$  avec  $a \neq 1$  alors nous trouvons  $c_0 = \frac{a}{a-1}$ . Soit donc

une série  $\sum_{n=1}^{\infty} \frac{1}{n^2} = 1,64493\dots$  c'est le  $\rho_2$ -algorithme qui correspond à

$c_0 = 2$ .

Comparons l' $\epsilon_2$ -algorithme et le  $\rho_2$ -algorithme

n	$S_n$	$\epsilon_2^{(n)}$	$\rho_2^{(n)}$
5	1,49	1,55	1,645
10	1,55	1,59	1,645
15	1,58	1,61	1,64495
20	1,59	1,620	1,64494
25	1,60	1,625	1,64493
30	1,61	1,628	1,64493

Nous pouvons remarquer que le  $\rho_2$ -algorithme accélère la convergence alors que  $\epsilon_2$  ne fait qu'améliorer la convergence.

Cependant si nous ne connaissons pas  $P$  ou si le calcul de  $c_0$  s'avère difficile on peut remplacer  $c_0$  par une approximation  $c_0^{(n)} = \frac{1}{1 + \Delta(\frac{1}{e^n})}$ .  
 Nous pouvons exprimer ce  $c_0^{(n)}$  en fonction de  $S_n$ ,  $S_{n+1}$  et  $S_{n+2}$ .

$$c_0^{(n)} = \frac{1}{1 + \frac{1}{\frac{\Delta S_{n+2}}{\Delta S_{n+1}} - 1} - \frac{1}{\frac{\Delta S_{n+1}}{\Delta S_n} - 1}} = \frac{1}{1 + \frac{\Delta S_{n+1}}{\Delta^2 S_{n+1}} - \frac{\Delta S_n}{\Delta^2 S_n}}$$

Si je remplace maintenant dans l'algorithme  $b_n$  par 0,  $b_0^{(n)}$  par 1 et  $c_0^{(n)}$  par cette valeur

j'obtiens :

$$\theta_2^{(n)} = \frac{S_{n+1} \Delta S_{n+2} \Delta^2 S_n - S_{n+2} \Delta S_n \Delta^2 S_{n+1}}{\Delta S_{n+2} \Delta^2 S_n - \Delta^2 S_{n+1} \Delta S_n}$$

nous retrouvons donc la 1ère étape d'algorithme que Brezinski appelle le  $\theta$ -algorithme et qui a également été obtenu par Germain-Bonne [26] en effectuant une extrapolation linéaire à partir des suites  $\{S_n\}$  et  $\{\epsilon_2^{(n)}\}$ . Il faut cependant remarquer que cette algorithme n'est plus un algorithme de losange en effet le calcul de  $\theta_2^{(n)}$  nécessite quatre termes de la suite initiale.

Théorème 17 : si  $\lim_{n \rightarrow \infty} \theta_2^{(n)} = \lim_{n \rightarrow \infty} S_n = S$ , alors une condition nécessaire et suffisante pour que  $\theta_2^{(n)}$  converge plus vite que  $S_{n+1}$  est que  $\lim_{n \rightarrow \infty} c_0^{(n)}$  existe

Corollaire : si  $\lim_{n \rightarrow \infty} \theta_2^{(n)} = \lim_{n \rightarrow \infty} S_n = S$ , alors une condition nécessaire et suffisante pour que  $\theta_2^{(n)}$  converge plus vite que  $S_{n+1}$  est que  $\lim_{n \rightarrow \infty} \Delta\left(\frac{\Delta S_n}{\Delta^2 S_n}\right)$  existe.

Exemple : pour les trois exemples précédents ce nouvel algorithme avec  $b_0^{(n)} = 1$  et  $c_0^{(n)} = \frac{1}{1 + \Delta\left(\frac{1}{e_n}\right)}$  nous donne

n	ex.1		ex.2	ex.3
0	0,933		0,99...	1,1388
5	- 0,211	$10^{-2}$	16 chiffres exacts	1,64461
10	- 0,209	$10^{-2}$	"	1,6447
15	- 0,193	$10^{-2}$	"	1,64491
20	- 0,185	$10^{-2}$	"	1,64492
25	- 0,177	$10^{-2}$	"	1,64493

### II.3.3 APPLICATION REPETEE DES ALGORITHMES DE TYPE $\theta_2$

Il apparaît intéressant d'étudier l'application répétée des algorithmes de type  $\theta_2$ . Dans le cas de l' $\epsilon$ -algorithme, cette première étape s'identifie au  $\Delta^2$  d'Aitken et nous savons que la mise en oeuvre de ce procédé n'est pas dépourvue d'efficacité [50,53]. Pour cette étude nous allons regarder le comportement asymptotique de  $\Delta\theta_1^{(n)}$  par rapport à  $\Delta S_n$  et  $\Delta\theta_2^{(n)}$ . Ces résultats débouchent sur des théorèmes d'accélération de convergence et des conditions d'application.

Théorème 18 : si l'on applique le  $\theta_2(b_n, a_k^{(n)})$  algorithme à une suite de nombres réels  $\{S_n\}$  avec  $a_0^{(n)} \neq 0 \forall n$

- si la condition de régularité  $A_0 + b_0 - K_0 \neq 0$  est vérifiée

$$\text{alors } \lim_{n \rightarrow \infty} \frac{\Delta\theta_1^{(n+1)}}{\Delta\theta_1^{(n)}} = \frac{b_0}{K_0} \quad (\text{où } K_0 \neq 0)$$

- si de plus  $c_0 = (K_0 - b_0 - A_0) / (K_0 - 1)$  avec  $K_0 \neq 1$  alors

$$\begin{aligned} \cdot \lim_{n \rightarrow \infty} \frac{\Delta\theta_1^{(n)} \Delta S_n}{a_1^{(n)}} &= \frac{1 - K_0}{K_0} \\ \cdot \lim_n \frac{\Delta\theta_2^{(n)} \Delta\theta_1^{(n)}}{a_1^{(n)}} &= 0 \end{aligned}$$

Démonstration : si nous avons la condition de régularité

$$\frac{\Delta\theta_1^{(n+1)}}{\Delta\theta_1^{(n)}} = \frac{\Delta b_{n+2} + \frac{a_o^{(n+2)}}{\Delta S_{n+2}} - \frac{a_o^{(n+1)}}{\Delta S_{n+1}}}{\Delta b_{n+1} + \frac{a_o^{(n+1)}}{\Delta S_{n+1}} - \frac{a_o^{(n)}}{\Delta S_n}} \quad \text{puisque } \left. \begin{array}{l} a_o^{(n)} \neq 0 \\ S_n \neq 0 \end{array} \right\} \forall n$$

$$\frac{\Delta\theta_1^{(n+1)}}{\Delta\theta_1^{(n)}} = \left[ \frac{\frac{\Delta b_{n+2}}{a_o^{(n+1)}} + \frac{\Delta S_{n+2}}{a_o^{(n+1)}} - \frac{\Delta S_{n+2}}{\Delta S_{n+1}}}{\frac{\Delta b_{n+1}}{a_o^{(n+1)}} + \frac{a_o^{(n+1)}}{a_o^{(n)}} - \frac{\Delta S_{n+1}}{\Delta S_n}} \right] \frac{\Delta S_{n+1}}{\Delta S_{n+2}} \times \frac{a_o^{(n+1)}}{a_o^{(n)}}$$

$$\lim_{n \rightarrow \infty} \frac{\Delta\theta_1^{(n+1)}}{\Delta\theta_1^{(n)}} = \frac{A_o + b_o - K_o}{A_o + b_o - K_o} \times \frac{b_o}{K_o} = \frac{b_o}{K_o}$$

car  $A_o + b_o - K_o \neq 0$

$$* \frac{\Delta\theta_1^{(n)}}{a_1^{(n)}} \frac{\Delta S_n}{a_1} = \left( \Delta b_{n+1} + \frac{a_o^{(n+1)}}{\Delta S_{n+1}} - \frac{a_o^{(n)}}{\Delta S_n} \right) \frac{\Delta S_n}{a_1}$$

$$\frac{\Delta\theta_1^{(n)}}{a_1^{(n)}} \Delta S_n = \frac{a_o^{(n)}}{a_1^{(n)}} \left[ \frac{\Delta S_n \Delta b_{n+1}}{a_o^{(n)}} + \frac{a_o^{(n+1)}}{a_o^{(n)}} \frac{\Delta S_n}{\Delta S_{n+1}} - 1 \right]$$



puisque  $\lim_{n \rightarrow \infty} c_o^{(n)} = c_o = \frac{A_o + b_o - K_o}{1 - K_o}$  pour  $K_o \neq 1$

$$\Rightarrow \lim_{n \rightarrow \infty} \frac{\Delta \theta_1^{(n)} \Delta S_n}{a_1^{(n)}} = \frac{1 - K_o}{A + b_o - K_o} \frac{A_o + b_o - K_o}{K_o} \quad \text{puisque } A + b_o - K_o \neq 0$$

$$\Rightarrow \lim_{n \rightarrow \infty} \frac{\Delta \theta_1^{(n)} \Delta S_n}{a_1^{(n)}} = \frac{1 - K_o}{K_o}$$

$$\frac{\Delta \theta_2^{(n)} \Delta \theta_1^{(n)}}{a_1^{(n)}} = \frac{\Delta \theta_1^{(n)}}{a_1^{(n)}} \left[ \Delta S_{n+1} + \frac{a_1^{(n+1)}}{\Delta \theta_1^{(n+1)}} - \frac{a_1^{(n)}}{\Delta \theta_1^{(n)}} \right]$$

nous pouvons remarquer que si  $c_o \neq 0, \infty : \lim_{n \rightarrow \infty} \frac{a_1^{(n+1)}}{a_1^{(n)}} = b_o$

$$\frac{\Delta \theta_2^{(n)} \Delta \theta_1^{(n)}}{a_1^{(n)}} = \frac{\Delta S_{n+1} \Delta \theta_1^{(n)}}{a_1^{(n)}} + \frac{a_1^{(n+1)}}{a_1^{(n)}} \frac{\Delta \theta_1^{(n)}}{\Delta \theta_1^{(n+1)}} - 1$$

$$\text{d'où } \lim_{n \rightarrow \infty} \frac{\Delta \theta_2^{(n)} \Delta \theta_1^{(n)}}{a_1^{(n)}} = 1 - K_o + \frac{b_o K_o}{b_o} - 1 = 0$$

C.Q.F.D.

Dans ce théorème nous pouvons remarquer que si  $S_n$  converge et si le  $\theta_2$ -algorithme est régulier alors  $\theta_1^{(n)}$  diverge.

Théorème 19 : si nous appliquons le  $\theta_2(b_n, a_k^{(n)})$ -algorithme à une suite  $\{S_n\}$  de limite  $S$  si  $c_o \neq 0$  et  $A_o + b_o - K_o \neq 0$  alors une condition nécessaire et suffisante pour que  $(\theta_2^{(n)} - S) = o(S_{n+1} - S)$  est que

$$\lim_{n \rightarrow \infty} \frac{\Delta \theta_2^{(n)} \Delta \theta_1^{(n)}}{a_1^{(n)}} = 0.$$

Démonstration : il nous reste à démontrer la condition nécessaire.

$$\text{Si } \lim_{n \rightarrow \infty} \frac{\Delta \theta_2^{(n)} \Delta \theta_1^{(n)}}{a_1^{(n)}} = 0$$

$$\frac{\Delta \theta_2^{(n)} \Delta \theta_1^{(n)}}{a_1^{(n)}} = \frac{\Delta S_{n+1} \Delta \theta_1^{(n)}}{a_1^{(n)}} + \frac{a_1^{(n+1)}}{a_1^{(n)}} \frac{\Delta \theta_1^{(n)}}{\Delta \theta_1^{(n+1)}} - 1$$

$$\text{or } c_0 \neq 0 \Rightarrow \lim_{n \rightarrow \infty} \frac{\Delta \theta_2^{(n)} \Delta \theta_1^{(n)}}{a_1^{(n)}} = \lim_{n \rightarrow \infty} \frac{\Delta S_{n+1} \Delta \theta_1^{(n)}}{a_1^{(n)}} + K_0 - 1$$

$$\frac{\Delta S_{n+1} \Delta \theta_1^{(n)}}{a_1^{(n)}} = \frac{a_0^{(n)}}{a_1^{(n)}} \frac{\Delta S_{n+1} b_{n+1}}{a_0^{(n)}} + \frac{a_0^{(n+1)}}{a_0^{(n)}} \frac{\Delta S_{n+1}}{\Delta S_{n+1}} - \frac{\Delta S_{n+1}}{\Delta S_n}$$

$$\Rightarrow \lim_{n \rightarrow \infty} \frac{\Delta \theta_2^{(n)} \Delta \theta_1^{(n)}}{a_1^{(n)}} = \frac{1}{c_0} A_0 + b_0 - K_0 + K_0 - 1 = 0$$

$$\left. \begin{array}{l} \text{or } A + b_0 - K_0 \neq 0 \\ \text{et } K_0 - 1 \neq 0 \end{array} \right\} \Rightarrow c_0 = \frac{A_0 + b_0 - K_0}{1 - K_0} \text{ qui est la condition d'accélération}$$

C.Q.F.D.

Dans toute cette étude nous différencions le cas  $K_0 = 1$  et  $K_0 \neq 1$  il semble donc intéressant de connaître la limite de  $\Delta \theta_2^{(n+1)} / \Delta \theta_2^{(n)}$  ou simplement de savoir si cette limite est égale ou différente de 1.

De manière générale nous obtenons

$$\frac{\Delta\theta_2^{(n+1)}}{\Delta\theta_2^{(n)}} = \frac{\Delta S_{n+2} + \frac{a_1^{(n+2)}}{\Delta\theta_1^{(n+2)}} - \frac{a_1^{(n+1)}}{\Delta\theta_1^{(n+1)}}}{\Delta S_{n+1} + \frac{a_1^{(n+1)}}{\Delta\theta_1^{(n+1)}} - \frac{a_1^{(n)}}{\Delta\theta_1^{(n)}}}$$

si  $a_1^{(n)} \neq 0 \forall n$  nous avons

$$\frac{\Delta\theta_1^{(n+1)}}{\Delta\theta_1^{(n+2)}} \frac{a_1^{(n+1)}}{a_1^{(n)}} \left[ \frac{\Delta S_{n+2} \Delta\theta_1^{(n+2)}}{a_1^{(n+1)}} + \frac{a_1^{(n+2)}}{a_1^{(n+1)}} - \frac{\Delta\theta_1^{(n+2)}}{\Delta\theta_1^{(n+1)}}}{\frac{\Delta S_{n+1} \Delta\theta_1^{(n+1)}}{a_1^{(n)}} + \frac{a_1^{(n+1)}}{a_1^{(n)}} - \frac{\Delta\theta_1^{(n+1)}}{\Delta\theta_1^{(n)}}} \right]$$

nous obtenons à la limite l'indétermination  $\frac{0}{0}$ . Pour lever cette indétermination il nous faut faire des hypothèses supplémentaires H2 :

$$b_0^{(n)} = b_0 \quad \forall n$$

$$\frac{\Delta S_{n+1}}{\Delta S_n} = K_0 + e_n \quad \text{avec } \lim_{n \rightarrow \infty} e_n = 0$$

$$c_0^{(n)} = \frac{K_0 + e_n - b_0}{K_0 + e_n - 1} \Rightarrow c_0 = \frac{K_0 - b_0}{K_0 - 1}$$

$$b_n = c \quad \forall n \Rightarrow A_0^{(n)} = 0$$

Nous pouvons alors énoncer le résultat suivant :

Théorème 20 : soit  $\{S_n\}$  une suite de nombres réels. Si nous lui appliquons le  $\theta_2(c, a_k^{(n)})$ -algorithme vérifiant les hypothèses H2 et si  $-1 < K_0 < 1$   $K_0 \neq 0$  alors

$$- \lim_{n \rightarrow \infty} \frac{\Delta e_{n+1}}{\Delta e_n} = B \implies \lim_{n \rightarrow \infty} \frac{\Delta \theta_2^{(n+1)}}{\Delta \theta_2^{(n)}} = K_0 B$$

et donc  $-1 < K_0 B < 1$

Ce théorème nous donne une condition suffisante pour itérer le  $\theta_2$ -algorithme.

## II.4 ALGORITHME DE TYPE $\theta$ GÉNÉRALISATION AUX AUTRES COLONNES

La remarque faite à la fin du paragraphe II.2 nous dit que le passage de la suite  $\{\theta_{2k}^{(n)}\}$  à la suite  $\{\theta_{2k+2}^{(n)}\}$  peut être considéré comme un cas particulier des algorithmes de type  $\theta_2$ . Nous pouvons ainsi généraliser les théorèmes démontrés pour les algorithmes de type  $\theta_2$  aux autres colonnes du tableau .

Pour cela nous prendrons les notations suivantes

$$b_k^{(n)} = \frac{a_{2k}^{(n+1)}}{a_{2k}^{(n)}}$$

$$c_k^{(n)} = \frac{a_{2k+2}^{(n)}}{a_{2k}^{(n)}}$$

$$A_k^{(n)} = \frac{\Delta \theta_{2k-1}^{(n+1)} \Delta \theta_{2k}^{(n+1)}}{a_{2k}^{(n)}}$$

et nous nous placerons dans les hypothèses H3 :

$$\lim_{n \rightarrow \infty} b_k^{(n)} = b_k \neq 0 \text{ fini} \quad \forall k \geq 0$$

$$\lim_{n \rightarrow \infty} c_k^{(n)} = c_k \neq 0 \text{ et fini} \quad \forall k \geq 0$$

$$\lim_{n \rightarrow \infty} \frac{\theta_{2k}^{(n+1)} - S}{\theta_{2k}^{(n)} - S} = K_k \text{ si } K_k \neq 1 \implies \lim_{n \rightarrow \infty} \frac{\Delta \theta_{2k}^{(n+1)}}{\Delta \theta_{2k}^{(n)}} = K_k$$

$$\lim_{n \rightarrow \infty} A_k^{(n)} = A_k$$

Nous pouvons énoncer un premier résultat :

Théorème 21 : si nous appliquons le  $\theta(b_n, a_k^{(n)})$  à une suite  $\{S_n\}$  de limite  $S$  alors une condition nécessaire et suffisante pour que

$$\lim_{n \rightarrow \infty} \theta_{2k+2}^{(n)} = \lim_{n \rightarrow \infty} \theta_{2k}^{(n)} \text{ est que :}$$

$$\lim_{n \rightarrow \infty} \frac{a_{2k+1}^{(n)}}{\Delta \theta_{2k+1}^{(n)}} = 0.$$

Comme pour les algorithmes de type  $\theta_2$  ce résultat n'est pas facilement utilisable. Aussi nous donnons une autre condition de régularité qui n'est qu'une condition suffisante :

Théorème 22 : si nous appliquons le  $\theta(b_n, a_k^{(n)})$  algorithmes à une suite  $\{S_n\}$  de limite  $S$ . Alors

$$K_j - b_j - A_j \neq 0 \implies \lim_{n \rightarrow \infty} \theta_{2j+2}^{(n)} = \lim_{n \rightarrow \infty} \theta_{2j}^{(n)}$$

Démonstration :

$$\theta_{2j+2}^{(n)} = \theta_{2j}^{(n+1)} + \frac{a_{2j+1}^{(n)}}{\Delta\theta_{2j+1}^{(n)}} = \theta_{2j}^{(n+1)} + \frac{a_{2j}^{(n)}}{\Delta\theta_{2j-1}^{(n+1)} + \frac{a_{2j}^{(n+1)}}{\Delta\theta_{2j}^{(n+1)}} - \frac{a_{2j}^{(n)}}{\Delta\theta_{2j}^{(n)}}}$$

$a_{2j}^{(n)} \neq 0 \forall n$  donc

$$\theta_{2j+2}^{(n)} = \theta_{2j}^{(n+1)} + \frac{\frac{a_{2j+1}^{(n)}}{a_{2j}^{(n)}} \Delta\theta_{2j}^{(n+1)}}{\frac{\Delta\theta_{2j-1}^{(n+1)} \Delta\theta_{2j}^{(n+1)}}{a_{2j}^{(n)}} + \frac{a_{2j}^{(n+1)}}{a_{2j}^{(n)}} - \frac{\Delta\theta_{2j}^{(n+1)}}{\Delta\theta_{2j}^{(n)}}}$$

d'où la condition

C.Q.F.D.

Nous pouvons donner un important corollaire.

Corollaire : si nous appliquons le  $\theta(b_n, a_k^{(n)})$  algorithme à une

suite  $\{S_n\}$  de limite  $S$ . Si  $\{\frac{a_{2j-1}^{(n)}}{a_{2j}^{(n)}}\}$  est bornée et si  $\{\theta_{2j}^{(n)}\}$  converge plus vite que  $\{\theta_{2j-2}^{(n+1)}\}$  alors :

$$K_j - b_j \neq 0 \Rightarrow \lim_{n \rightarrow \infty} \theta_{2j+2}^{(n)} = \lim_{n \rightarrow \infty} \theta_{2j}^{(n)}.$$

Démonstration : en effet la condition du théorème 22 est  $K_j - b_j - A_j \neq 0$  or si  $\{\theta_{2j}^{(n)}\}$  converge plus vite que  $\{\theta_{2j-2}^{(n+1)}\}$  le théorème 18 nous dit que  $\frac{\Delta\theta_{2j}^{(n+1)}}{a_{2j-1}^{(n)}} \frac{\Delta\theta_{2j-1}^{(n+1)}}{n^{\infty}} \rightarrow 0$  et si  $\left\{\frac{a_{2j-1}^{(n)}}{a^{(n)}}\right\}$  est borné  $A_j = 0$  donc la condition s'écrit  $K_j - b_j \neq 0$

C.Q.F.D.

Maintenant nous allons généraliser les résultats d'accélération de la convergence aux autres colonnes du tableau  $\theta$ . Là encore nous séparons le cas ou  $K_j \neq 1 \forall j$  et le cas des suites à convergence logarithmique ou  $K_j = 1$ .

Théorème 23 : si nous appliquons le  $\theta(b_n, a_k^{(n)})$  algorithme à la suite  $\{S_n\}$ . Si  $K_j \neq 1$  alors

$$c_j = \frac{K_j - b_j - A_j}{K_j - 1} \iff \Delta\theta_{2j+2}^{(n)} = o(\Delta\theta_{2j}^{(n+1)})$$

Ce théorème est une application directe du théorème 13. De plus si la suite  $\{S_n\}$  converge vers  $S$  nous pouvons dire :

$$\left. \begin{array}{l} K_i \neq 1 \quad 0 \leq i \leq j \\ K_i - b_i - A_i \neq 0 \quad \forall i \text{ tel que } 0 \leq i \leq j \end{array} \right\} \implies \theta_{2j+2}^{(n)} - S = o(\theta_{2j}^{(n+1)} - S)$$

$$c_j = \frac{K_j - b_j - A_j}{K_j - 1}$$

Nous pouvons tirer du théorème 23 un important corollaire.

Corollaire : si nous appliquons le  $\theta(b_n, a_k^{(n)})$  algorithme à la suite

$\{\theta_n^{(n)}\}$ . Si  $K_j \neq 1$ , si  $\{\frac{a_{2j-1}^{(n)}}{a_{2j}^{(n)}}\}$  est bornée et si  $\{\theta_{2j}^{(n)}\}$  converge plus vite

que  $\{\theta_{2j-2}^{(n)}\}$  alors :

$$c_j = \frac{K_j - b_j}{K_j - 1} \iff \Delta\theta_{2j+2}^{(n)} = o(\Delta\theta_{2j}^{(n+1)})$$

Démonstration :

$$\left. \begin{array}{l} \{\theta_{2j}^{(n)}\} \text{ converge plus vite que } \{\theta_{2j-2}^{(n)}\} \\ \left\{ \frac{a_{2j-1}^{(n)}}{a_{2j}^{(n)}} \right\} \text{ bornée} \end{array} \right\} \implies A_j = 0$$

donc la condition s'écrit  $c_j = \frac{K_j - b_j}{K_j - 1}$

C.Q.F.D.

Nous pouvons faire la même remarque que pour les algorithmes de type  $\theta_2$ . En effet nous comparons  $\Delta\theta_{2j+2}^{(n)}$  et  $\Delta\theta_{2j}^{(n+1)}$  alors qu'il nous faut 3 termes  $\theta_{2j}^{(n)}$ ,  $\theta_{2j}^{(n+1)}$  et  $\theta_{2j}^{(n+2)}$  pour calculer  $\theta_{2j+2}^{(n)}$  ; cependant nous pouvons affirmer que si  $K_j \neq 0$  alors :

$$\Delta\theta_{2j+2}^{(n)} = o(\Delta\theta_{2j}^{(n+1)}) \implies \Delta\theta_{2j+2}^{(n)} = o(\Delta\theta_{2j}^{(n+2)}).$$

La difficulté majeure pour ce type d'algorithme est la connaissance des  $K_i$  et principalement nous devons avoir  $K_i \neq 1$ .

Bien que nous ne puissions vérifier l'hypothèse  $K_i \neq 1$  pour chaque  $i$  voici des exemples qui nous montre la puissance de ce type d'algorithme. Lorsque  $K_0 \neq 1$  nous prendrons  $c_i^{(n)} = b_i^{(n)} = 1$

$$1. \quad S_n = \sum_{i=1}^n \frac{(-1)^i}{2^i - 1} \quad S = 0,785398163397448$$

$2n+1$	$S_{n+1}$	$\epsilon_n^{(o)}$
5	0,834	0,7855
9	0,813	0,7853983
13	0,8046	0,7853981635
17	0,800	0,785398163395
21	0,7972	0,785398163397448

$$2. \quad \text{Ou } S_n \text{ est divergente } S_n = \sum_{i=1}^n (-2)^i / (i+1)$$

$$S = \text{Log}(3) = 1,098612288668$$

$2n+1$	$S_{2n+1}$	$10^{\frac{(o)}{2n}}$
5	0,506	$10^1$ 1,101
9	0,375	$10^2$ 1,09862
13	0,404	$10^3$ 1,0986123
17	0,503	$10^4$ 1,098612289
21	0,655	$10^5$ 1,098612288669

## II.5 STABILITÉ NUMÉRIQUE ET PROPAGATION DES ERREURS.

L'analyse des erreurs numériques sur ordinateur d'un procédé récursif peut être approchée de diverses manières. En faisant de fortes hypothèses concernant la façon dont les erreurs sont introduites il est possible d'obtenir de bonnes estimations de l'erreur et de cette manière d'imaginer un procédé de correction des quantités obtenues. Ceci a été fait pour l' $\epsilon$ -algorithme dans [56]. De même, en faisant certaines hypothèses concernant la façon dont sont introduites les erreurs, il est parfois possible d'utiliser la théorie algorithmique pour diagnostiquer les conditions pour lesquelles le calcul numérique est instable. Ceci se rencontre par exemple dans le cas bien connu de récurrence linéaire à trois termes : ce qui a été mis en oeuvre pour l' $\epsilon$ -algorithme dans [57]. Enfin il n'a pas été possible d'exploiter la théorie de l'algorithme et il nous faut, en adoptant une technique de perturbation linéarisée, se contenter d'un procédé qui donne une borne supérieure de l'erreur : ce qui est abordé pour l' $\epsilon$ -algorithme dans [58].

Dans l'investigation présente il a été possible d'étudier le comportement de l'erreur à l'aide du comportement asymptotique des quantités calculées.

Supposons que, dans le tableau  $\theta$ , nous introduisons une erreur relative  $\delta_{k-1}^{(n)}$  sur l'élément  $\theta_{k-1}^{(n)}$  ; si nous notons  $\delta_k^{(n-1)}$ ,  $\delta_k^{(n)}$ ,  $\delta_{k+1}^{(n-2)}$ ,  $\delta_{k+1}^{(n-1)}$  et  $\delta_{k+1}^{(n)}$ , les erreurs relatives respectives de  $\theta_k^{(n-1)}$ ,  $\theta_k^{(n)}$ ,  $\theta_{k+1}^{(n-2)}$ ,  $\theta_{k+1}^{(n-1)}$  et  $\theta_{k+1}^{(n)}$ , les erreurs se propagent dans le tableau  $\theta$  de la manière suivante :

$$\begin{array}{ccc}
 & \theta_k^{(n-2)} & \\
 & & \theta_{k+1}^{(n-2)} (1 + \delta_{k+1}^{(n-2)}) \\
 \theta_{k-1}^{(n-1)} & & \\
 \theta_{k-2}^{(n)} & \theta_k^{(n-1)} (1 + \delta_k^{(n-1)}) & \\
 & & \theta_{k+1}^{(n-1)} (1 + \delta_{k+1}^{(n-1)}) \\
 \theta_{k-1}^{(n)} (1 + \delta_{k-1}^{(n)}) & & \\
 \theta_{k-2}^{(n+1)} & \theta_k^{(n)} (1 + \delta_k^{(n)}) & \\
 & & \theta_{k+1}^{(n)} (1 + \delta_{k+1}^{(n)}) \\
 \theta_{k-1}^{(n+1)} & & \\
 & \theta_k^{(n+1)} & \\
 & & \theta_{k+1}^{(n+1)}
 \end{array}$$

Si l'erreur  $\delta_{k-1}^{(n)}$  est suffisamment petite nous pouvons obtenir certaines relations entre les erreurs :

$$\theta_k^{(n-1)} (1 + \delta_k^{(n-1)}) = \theta_{k-2}^{(n)} + \frac{a_{k-1}^{(n-1)}}{\theta_{k-1}^{(n)} (1 + \delta_{k-1}^{(n)}) - \theta_{k-1}^{(n-1)}}$$

$$\theta_k^{(n-1)} (1 + \delta_k^{(n-1)}) = \theta_{k-2}^{(n)} + \frac{\frac{a_{k-1}^{(n-1)}}{\Delta \theta_{k-1}^{(n-1)}}}{1 + \frac{\delta_{k-1}^{(n)} \theta_{k-1}^{(n)}}{\Delta \theta_{k-1}^{(n)}}}$$

et si  $\delta_{k-1}^{(n)}$  est suffisamment petit :

$$\theta_k^{(n-1)}(1+\delta_k^{(n-1)}) = \theta_{k-2} + \frac{a_{k-1}^{(n-1)}}{\Delta\theta_{k-1}^{(n-1)}} - \delta_{k-1}^{(n)} \frac{\theta_{k-1}^{(n)} a_{k-1}^{(n-1)}}{[\Delta\theta_{k-1}^{(n-1)}]^2}$$

d'où la relation

$$\delta_k^{(n-1)} = \delta_{k-1}^{(n)} a_{k-1}^{(n-1)} \frac{\theta_{k-1}^{(n)}}{\theta_k^{(n-1)} [\Delta\theta_{k-1}^{(n-1)}]^2}$$

en raisonnant de la même manière nous obtenons les relations :

$$\delta_k^{(n)} = \delta_{k-1}^{(n)} a_{k-1}^{(n)} \frac{\theta_{k-1}^{(n)}}{\theta_k^{(n)} [\Delta\theta_{k-1}^{(n)}]^2}$$

$$\delta_{k+1}^{(n-2)} = a_k^{(n-2)} a_{k-1}^{(n-1)} \frac{k-1}{\theta_{k+1}^{(n-2)} [\Delta\theta_{k-1}^{(n-1)}]^2 [\Delta\theta_k^{(n-2)}]^2} \delta_{k-1}^{(n)}$$

$$\delta_{k+1}^{(n)} = - a_k^{(n)} a_{k-1}^{(n)} \frac{k-1}{\theta_{k+1}^{(n)} [\Delta\theta_{k-1}^{(n)}]^2 [\Delta\theta_k^{(n)}]^2} \delta_{k-1}^{(n)}$$

$$\delta_{k+1}^{(n-1)} = \frac{\theta_{k-1}^{(n)}}{\theta_{k+1}^{(n-1)}} \left[ 1 - \left\{ \frac{a_{k-1}^{(n)}}{[\Delta\theta_{k-1}^{(n)}]^2} + \frac{a_{k-1}^{(n-1)}}{[\Delta\theta_{k-1}^{(n-1)}]^2} \right\} \frac{a_k^{(n-1)}}{[\Delta\theta_k^{(n-1)}]^2} \right] \delta_{k-1}^{(n)}$$

De même, nous nous intéressons aux erreurs absolues et leurs propagations dans le tableau  $\theta$ . Soit  $E_{k-1}^{(n)}$  l'erreur absolue faite sur l'élément  $\theta_{k-1}^{(n)}$ , si nous notons  $E_k^{(n-1)}$ ,  $E_k^{(n)}$ ,  $E_{k+1}^{(n-2)}$ ,  $E_{k+1}^{(n-1)}$  et  $E_{k+1}^{(n)}$  les erreurs absolues respectives de  $\theta_k^{(n-1)}$ ,  $\theta_k^{(n)}$ ,  $\theta_{k+1}^{(n-2)}$ ,  $\theta_{k+1}^{(n-1)}$ ,  $\theta_{k+1}^{(n)}$  et si  $E_{k-1}^{(n)}$  est suffisamment petite alors nous avons les relations suivantes :

$$E_k^{(n-1)} = - a_{k-1}^{(n-1)} \frac{E_{k-1}^{(n)}}{[\Delta\theta_{k-1}^{(n-1)}]^2}$$

$$E_k^{(n)} = a_{k-1}^{(n)} \frac{E_{k-1}^{(n)}}{[\Delta\theta_{k-1}^{(n)}]^2}$$

$$E_{k+1}^{(n-2)} = a_k^{(n-2)} a_{k-1}^{(n-1)} \frac{E_{k-1}^{(n)}}{[\Delta\theta_k^{(n-2)}]^2 [\Delta\theta_{k-1}^{(n-1)}]^2}$$

$$E_{k+1}^{(n-1)} = \left[ 1 - \left\{ \frac{a_{k-1}^{(n)}}{[\Delta\theta_{k-1}^{(n)}]^2} + \frac{a_{k-1}^{(n-1)}}{[\Delta\theta_{k-1}^{(n-1)}]^2} \right\} \frac{a_k^{(n-1)}}{[\Delta\theta_k^{(n-1)}]^2} \right] E_{k-1}^{(n)}$$

$$E_{k+1}^{(n)} = - a_k^{(n)} a_k^{(n)} \frac{E_{k-1}^{(n)}}{[\Delta\theta_{k-1}^{(n)}]^2 [\Delta\theta_k^{(n)}]^2}$$

Ce qui nous intéresse ici, ce sont les erreurs qui se rapportent aux colonnes paires c'est-à-dire le cas où  $k=2p+1$  dans les formules ci-dessus ces erreurs dépendent des rapports

$$a_k^{(n-2)} a_{k-1}^{(n-1)} \frac{1}{[\Delta\theta_k^{(n-2)} \Delta\theta_{k-1}^{(n-1)}]^2} \text{ et } a_k^{(n)} a_{k-1}^{(n-1)} \frac{1}{[\Delta\theta_{k-1}^{(n)} \Delta\theta_k^{(n)}]^2}$$

Certes nous ne pouvons pas calculer ces rapports, cependant si l'on connaît les  $K_j$ , il nous est possible de savoir comment se comporte ce rapport, lorsque  $n$  tend vers l'infini ceci à l'aide du théorème 18. En effet si  $\{\theta_{2j}^{(n)}\}$  converge plus vite que  $\{\theta_{2j-2}^{(n)}\}$  et si  $K_j \neq 1$  nous avons :

$$\lim_{n \rightarrow \infty} a_{2j+1}^{(n-2)} a_{2j}^{(n-1)} \frac{1}{[\Delta\theta_{2j+1}^{(n-2)} \cdot \Delta\theta_{2j+1}^{(n-1)}]^2} = \frac{b_j}{(b_j - K_j)(1 - K_j)}$$

$$\lim_{n \rightarrow \infty} a_{2j+1}^{(n)} a_{2j}^{(n-1)} \frac{1}{[\Delta\theta_{2j}^{(n)} \cdot \Delta\theta_{2j+1}^{(n)}]^2} = \frac{1}{b_j(b_j - K_j)} \cdot \frac{K_j^2}{(1 - K_j)}$$

Nous pouvons ainsi remarquer :

- si nous connaissons  $K_j$  il est possible de choisir  $b_j$  de telle manière que ces facteurs ne soient pas trop grands
- dans la pratique nous ne connaissons pas les  $K_j$  cependant nous pouvons dire que si ces facteurs sont petits la propagation de l'erreur sera moins sensible. C'est ce que nous avons mis en évidence sur les exemples suivants :

nous appliquons le  $\theta$  algorithme avec  $b_0^{(n)} = b_0 = 1$  et  $c_0^{(n)} = c_0 = 1$  dans le premier exemple  $K_0 > 0$  dans le second  $K_0 < 0$



$$S_n = 1 + \sum_{i=1}^{\infty} \left(\frac{1}{i+1}\right)^{n+2}$$

n	$S_n$	$\theta_2^{(n)}$	$\theta_4^{(n)}$	$\theta_6^{(n)}$
2	1,0823232337	1,0024825995		
3	1,0369277551	1,0007117685	1,0000443733	
4	1,0173430619	1,0002125026	1,0000089237	1,00000005006
5	1,0083492773	1,0000652737	1,0000018823	
6	1,0040773562	1,0000204666		
7	1,0020083928			
8	1,0009975751			

$$S_n = 1 + (-1)^n \sum_{i=1}^{\infty} \left(\frac{1}{i+1}\right)^{n+2}$$

n	$S_n$	$\theta_2^{(n)}$	$\theta_4^{(n)}$	$\theta_6^{(n)}$
2	1,0823232337	1,0003692783		
3	0,9630722448	0,9999057281	1,0000019547	
4	1,0173430619	1,0000263230	0,9999996609	1,00000000089
5	0,9916507226	0,9999922324	1,0000000649	
6	1,0040773561	1,0000023760		
7	0,9979916071			
8	1,0009945751			

Dans ces deux exemples nous avons  $K'_0 = -K_0$  et c'est pour le cas  $K'_0$  négatif que les meilleurs résultats sont obtenus. Pour les autres colonnes nous ne savons pas si  $K_j$  existe mais c'est pour une suite oscillante que les résultats sont les meilleurs.

## CHAPITRE III



## APPLICATIONS AUX ALGORITHMES CONNUS

Au chapitre précédent nous avons étudié les algorithmes de type  $\theta$  dans toute leur généralité. Nous avons obtenu ainsi des conditions sur les suites  $b_n, a_k^{(n)}$ , ce qui nous donne pour chaque suite une famille d'algorithmes capable d'accélérer sa convergence. Le but de ce chapitre est quelque peu différent : les algorithmes sont connus, c'est-à-dire les  $b_n$  et  $a_k^{(n)}$  fixés, et nous donnons des conditions sur la suite, pour que celle-ci soit accélérée par l'algorithme en question. Nous passons ainsi en revue un certain nombre d'algorithmes connus.

III.1 L' $\epsilon$ -ALGORITHME.

C'est à l'heure actuelle le plus puissant et le plus étudié des algorithmes d'accélération de la convergence. Nous pouvons le considérer comme une généralisation du  $\Delta^2$  d'Aitken.

Le  $\Delta^2$  d'Aitken consiste à transformer une suite  $\{S_n\}$  en une suite  $\{\epsilon_2^{(n)}\}$  donnée par :

$$\epsilon_2^{(n)} = \frac{S_{n+2} S_n - S_{n+1}^2}{S_{n+2} - 2S_{n+1} + S_n} = S_{n+1} - \frac{\Delta S_n \cdot \Delta S_{n+1}}{\Delta^2 S_n}$$

$$\epsilon_2^{(n)} = S_{n+1} - \frac{\Delta S_{n+1}}{\frac{\Delta S_{n+1}}{\Delta S_n} - 1}$$

qui peut encore se noter

$$\epsilon_2^{(n)} = \frac{\begin{vmatrix} S_n & S_{n+1} \\ \Delta S_n & \Delta S_{n+1} \end{vmatrix}}{\begin{vmatrix} 1 & 1 \\ \Delta S_n & \Delta S_{n+1} \end{vmatrix}}$$

Shank [50] a introduit et étudié une transformation non linéaire qui généralise le procédé  $\Delta^2$  d'Aitken. Cette transformation  $e_k(S_n)$  est définie de la manière suivante :

$$e_k(S_n) = \frac{\begin{vmatrix} S_n & \dots & S_{n+k} \\ S_{n+1} & \dots & S_{n+k+1} \\ \dots & \dots & \dots \\ S_{n+k} & \dots & S_{n+2k} \end{vmatrix}}{\begin{vmatrix} \Delta^2 S_n & \dots & \Delta^2 S_{n+k-1} \\ \dots & \dots & \dots \\ \Delta^2 S_{n+k+1} & \dots & \Delta^2 S_{n+2k-2} \end{vmatrix}}$$

L' $\epsilon$ -algorithme, dû à P. Wynn [55] est un algorithme récursif qui évite le calcul effectif de ces déterminants. Les règles de cet algorithme sont les suivantes :

$$\begin{aligned} \epsilon_{-1}^{(n)} &= 0 & \epsilon_0^{(n)} &= S_n & \forall n \in \mathbb{N} \\ \epsilon_{k+1}^{(n)} &= \epsilon_{k-1}^{(n+1)} + \frac{1}{\frac{\epsilon_k^{(n+1)}}{\epsilon_k^{(n)}} - \epsilon_k^{(n)}} & \forall k \in \mathbb{N} & \forall n \in \mathbb{N} \end{aligned}$$

nous avons alors :

$$\epsilon_{2k}^{(n)} = e_k(S_n)$$

Cet algorithme est un algorithme en losange déjà défini au chapitre précédent. Un fait très important également est la connection de l' $\epsilon$ -algorithme avec la table de Padé. Beaucoup de physiciens sont intéressés par l'approximation de Padé, car ils obtiennent souvent la solution à leurs problèmes sous la forme de série de puissances, chaque terme de cette série étant très difficile à calculer ils emploient l'approximation de Padé pour essayer d'améliorer leurs résultats et avoir le moins possible de termes à calculer. Nous allons voir qu'il y a identité entre l' $\epsilon$ -algorithme et la moitié supérieure de la table de Padé.

Etant donnée une série de puissances  $f(x) = \sum_{i=0}^{\infty} c_i x^i$  il est possible de construire, sous certaines conditions, une suite double de fonctions rationnelles de  $x$  qui seront notées  $p/q$  avec

$$[p/q] = \frac{\sum_{i=0}^p a_i x^i}{\sum_{i=0}^q b_i x^i}$$

$$\text{tel que } [p/q] = \sum_{i=0}^{p+q} c_i x^i + \sum_{i=p+q+1}^{\infty} d_i x^i$$

la fonction rationnelle  $[p/q]$  s'appelle approximant de padé de  $f(x)$ . La table à double entrée obtenue en plaçant  $p/q$  à l'intersection de la ligne  $q$  et la colonne  $p$  ( $q, p \in \mathbb{N}$ ) s'appelle la table de Padé [43].

De nombreux ouvrages ont été écrits sur la table de Padé. Les plus importants sont ceux de PERRON [45] et WALL [52] et plus récemment [28]. La connection entre la table de Padé et l' $\epsilon$ -algorithme a été mise en lumière par Shank [1] et Wynn [60].

Théorème 23 : soit  $f(x) = \sum_{i=0}^{\infty} c_i x^i$  une série de puissances. Si on applique l' $\epsilon$ -algorithme aux sommes partielles de cette série  $S_n = \sum_{i=0}^n c_i x^i$  alors :

$$\epsilon_{2k}^{(n)} = [n + k / K]$$

La connection entre l' $\epsilon$ -algorithme et les fractions continues a été mise en évidence par Wynn[61]. Enfin Brezinski a relié tout ceci à la théorie des polynomes orthogonaux ce qui lui a permis d'obtenir d'autres approximants rationnels que ceux de Padé.

Nous allons maintenant appliquer les résultats du chapitre II à l' $\epsilon$ -algorithme ainsi tous les résultats démontrés pour l' $\epsilon$ -algorithme s'appliqueront aux approximants de Padé et aux divers éléments connectés à l' $\epsilon$ -algorithme. Certains des résultats obtenus sont déjà connus mais d'autres résultats importants sont nouveaux en particulier ce qui concerne la convergence et l'accélération de la convergence pour lesquelles nous avons jusqu'à présent peu de résultats.

Nous retrouvons ainsi la règle de la croix : propriété 2 du chapitre II ainsi que le résultat suivant : bien que l' $\epsilon$ -algorithme soit un procédé non linéaire nous pouvons dire

Propriété 3 : si l'application de l' $\epsilon$ -algorithme à  $\{S_n\}$  et à  $\{aS_n + b\}$  fournit respectivement les quantités  $\epsilon_k^{(n)}$  et  $\bar{\epsilon}_k^{(n)}$  alors

$$\bar{\epsilon}_{2k}^{(n)} = a\epsilon_{2k}^{(n)} + b$$

$$\bar{\epsilon}_{2k+1}^{(n)} = \epsilon_{2k+1}^{(n)} / a$$

Nous allons, comme pour les autres algorithmes, étudier le cas particulier de la deuxième colonne soit le  $\Delta^2$  d'Aitken. Cet algorithme a été très étudié, en effet le  $\Delta^2$  d'Aitken est la première étape de nombreux algorithmes.

Théorème 24 : une condition nécessaire et suffisante pour que  $\epsilon_2^{(n)} = S$   $\forall n > N$  est que :

$$S_n = S + a \lambda^n \text{ avec } \lambda \neq 1$$

Démonstration : dans le théorème 10  $a_0^{(i)} = a_1^{(i)} = 1 \forall i$  et  $\rho_i = \rho_0$   
 $\Rightarrow S_n = S + A (1 + \rho_0)^n$

Remarque : on voit que  $\epsilon_2^{(n)} = S$  même si  $|\lambda| > 1$ , c'est-à-dire même si la suite  $\{S_n\}$  diverge,  $S$  est alors l'autre limite. Si  $|\lambda| < 1$  alors  $S$  est la limite de  $\{S_n\}$ . Si  $\lambda = -1$  il y a deux points d'accumulation,  $S$  est la moyenne de ces points d'accumulations.

Théorème 25 : si on applique le  $\Delta^2$  d'Aitken à une suite  $\{S_n\}$  qui converge vers  $S$  et si :

$$\lim_{n \rightarrow \infty} \frac{S_{n+1} - S}{S_n - S} = K_0 \neq 1 \text{ alors}$$

$\{\epsilon_2^{(n)}\}$  converge vers  $S$  plus vite que  $\{S_{n+1}\}$ .

Démonstration : dans les théorèmes 12 et 13 nous avons  $b_n = 0$

$a_0^{(n)} = a_1^{(n)} = 1$  ce qui se traduit par

$A_0^{(n)} = 0$   $b_0^{(n)} = 1$  et  $c_0^{(n)} = 1$

donc  $c_0 = \frac{K_0 - 1}{K_0 - 1} = 1 \Leftrightarrow \epsilon_2^{(n)} - S = 0 (S_{n+1} - S)$

C.Q.F.D.

Là également il n'est pas nécessaire que la suite converge pour parler d'accélération de la convergence :

Théorème 26 : si on applique le procédé  $\Delta^2$  d'Aitken à une suite  $\{S_n\}$  et si :

$$\lim_{n \rightarrow \infty} \frac{\Delta S_{n+1}}{\Delta S_n} = K_0 \neq 1 \text{ alors}$$

$$\Delta \epsilon_2^{(n)} = o(\Delta S_{n+1})$$

Démonstration : elle est évidente en appliquant le théorème 14.

Nous avons écarté dans ces théorèmes les suites à convergence logarithmique c'est-à-dire telles que :

$$\lim_{n \rightarrow \infty} \frac{S_{n+1} - S}{S_n - S} = 1$$

Cependant il est parfois possible d'accélérer la convergence de ce type de suite : nous nous plaçons d'abord dans l'hypothèse  $\frac{S_{n+1} - S}{S_n - S} = 1 + e_n$  avec  $e_n \xrightarrow{n \rightarrow \infty} 0$

Théorème 27 : si, dans les conditions précédentes,

$$\lim_{n \rightarrow \infty} \frac{\Delta S_{n+1}}{\Delta S_n} = \lim_{n \rightarrow \infty} \frac{e_{n+1}}{e_n} = a \neq 1 \text{ alors}$$

$$\lim_{n \rightarrow \infty} \epsilon_2^{(n)} = S$$

en ce qui concerne l'accélération de la convergence.

Théorème 28 : si  $\{\epsilon_2^{(n)}\}$  converge alors

$$\lim_{n \rightarrow \infty} \Delta\left(\frac{1}{e_n}\right) = 0 \iff (\epsilon_2^{(n)}) = 0 \quad (S_{n+1} - S)$$

Erreur

$$\epsilon_2^{(n)} - S = 0 \quad (S_{n+1} - S)$$

Démonstration : si la suite  $\{\epsilon_2^{(n)}\}$  converge, elle converge vers S

(Lubkin) et d'après le théorème 15 : une condition nécessaire et suffisante pour qu'il y ait accélération de la convergence est que :

$$c_0 = 1 = 1 + \lim_{n \rightarrow \infty} \Delta\left(\frac{1}{e_n}\right) \text{ d'où le résultat}$$

C.Q.F.D.

En général nous ne connaissons pas S dans ce cas les hypothèses sont faites sur le rapport  $\frac{\Delta S_{n+1}}{\Delta S_n}$ . Nous nous plaçons donc dans l'hypothèse suivante :

$$\frac{\Delta S_{n+1}}{\Delta S_n} = 1 + d_n \text{ avec } \lim_{n \rightarrow \infty} d_n = 0$$

Théorème 29 : soit  $S_n$  une suite vérifiant l'hypothèse précédente

$$\lim_{n \rightarrow \infty} \frac{1}{1 + \Delta\left(\frac{1}{d_n}\right)} = 1 \iff \Delta \epsilon_2^{(n)} = 0 \quad (\Delta S_{n+1}) \text{ de plus}$$

$$\lim_{n \rightarrow \infty} \frac{\Delta S_n \Delta S_{n+1}}{\Delta^2 S_n} = 0 \implies \lim_{n \rightarrow \infty} \epsilon_2^{(n)} = \lim_{n \rightarrow \infty} S_n$$

Démonstration : il suffit d'appliquer le théorème 16 au cas

$$b_0^{(n)} = 1, b_n = 0 \text{ et } c_0^{(n)} = 1 \quad \forall n \in \mathbb{N}$$

Nous pouvons également noter une nouvelle écriture du  $\Delta^2$  d'Aitken :

$$\epsilon_2^{(n)} = \frac{\Delta\left(\frac{S_n}{\Delta S_n}\right)}{\Delta\left(\frac{1}{\Delta S_n}\right)}$$

Nous voyons que la seule condition est  $\lim_{n \rightarrow \infty} \frac{\Delta S_{n+1}}{\Delta S_n} \neq 1$ . Il paraît donc intéressant d'étudier

$$\lim_{n \rightarrow \infty} \frac{\Delta \epsilon_2^{(n+1)}}{\Delta \epsilon_2^{(n)}} \text{ et en particulier le cas où : } \lim_{n \rightarrow \infty} \frac{\Delta \epsilon_2^{(n+1)}}{\Delta \epsilon_2^{(n)}} \neq 1, 0$$

ceci afin d'étudier le  $\Delta^2$  d'Aitken itéré :

Théorème 30 : si  $-\frac{\Delta S_{n+1}}{\Delta S_n} = K_0 + e_n$  avec  $K_0 \neq 1, 0$  et  $\lim_{n \rightarrow \infty} e_n = 0$

$$- \lim_{n \rightarrow \infty} \frac{\Delta e_{n+1}}{\Delta e_n} \text{ existe}$$

Alors

$$\lim_{n \rightarrow \infty} \frac{\Delta \epsilon_2^{(n+1)}}{\Delta \epsilon_2^{(n)}} = K_0 \lim_{n \rightarrow \infty} \frac{\Delta e_{n+1}}{\Delta e_n} \text{ et } \lim_{n \rightarrow \infty} \left| \frac{\Delta e_{n+1}}{\Delta e_n} \right| \leq 1$$

Démonstration :

$$\frac{\Delta \epsilon_2^{(n+1)}}{\Delta \epsilon_2^{(n)}} = \frac{\Delta S_{n+2} \left[ \frac{1}{\Delta \epsilon_1^{(n+2)}} - \frac{1}{\Delta \epsilon_1^{(n+1)}} \right]}{\Delta S_{n+1} \left[ \frac{1}{\Delta \epsilon_1^{(n+1)}} - \frac{1}{\Delta \epsilon_1^{(n)}} \right]} = \frac{\Delta \epsilon_1^{(n+1)}}{\Delta \epsilon_1^{(n+2)}} \left[ \frac{\Delta S_{n+2} \Delta \epsilon_1^{(n+2)} + 1 - \frac{\Delta \epsilon_1^{(n+2)}}{\Delta \epsilon_1^{(n+1)}}}{\Delta S_{n+1} \Delta \epsilon_1^{(n+1)} + 1 - \frac{\Delta \epsilon_1^{(n+1)}}{\Delta \epsilon_1^{(n)}}} \right]$$

nous allons d'abord étudier :

$$\Delta S_n \Delta \epsilon_1^{(n)} = \Delta S_n \left[ \frac{1}{\Delta S_{n+1}} - \frac{1}{\Delta S_n} \right] = \frac{\Delta S_n}{\Delta S_{n+1}} - 1$$

avec l'hypothèse faite :

$$\Delta S_n \Delta \epsilon_1^{(n)} = \frac{1}{K_0 + e_n} - 1$$

Etudions maintenant

$$\frac{\Delta \epsilon_1^{(n+1)}}{\Delta \epsilon_1^{(n)}} = \frac{-\frac{1}{\Delta S_{n+2}} - \frac{1}{\Delta S_{n+1}}}{\frac{1}{\Delta S_{n+1}} - \frac{1}{\Delta S_n}} = \frac{\Delta S_{n+1}}{\Delta S_{n+2}} \cdot \frac{1 - \frac{\Delta S_{n+2}}{\Delta S_{n+1}}}{1 - \frac{\Delta S_{n+1}}{\Delta S_n}}$$

$$\frac{\Delta \epsilon_1^{(n+1)}}{\Delta \epsilon_1^{(n)}} = \frac{1}{K_o + e_{n+1}} \cdot \frac{1 - K_o - e_{n+1}}{1 - K_o - e_n}$$

d'où

$$\frac{\Delta \epsilon_2^{(n+1)}}{\Delta \epsilon_2^{(n)}} = (K_o + e_{n+2}) \cdot \frac{1 - K_o - e_{n+1}}{1 - K_o - e_{n+2}} \cdot \left[ \frac{\frac{1}{K_o + e_{n+2}} - 1 + 1 - \frac{1}{K_o + e_{n+2}}}{\frac{1}{K_o + e_{n+1}} - 1 + 1 - \frac{1}{K_o + e_{n+1}}} \cdot \frac{1 - K_o - e_{n+2}}{1 - K_o - e_{n+1}} \right]$$

$$\frac{\Delta \epsilon_2^{(n+1)}}{\Delta \epsilon_2^{(n)}} = (K_o + e_{n+2}) \cdot \frac{1 - K_o - e_{n+1}}{1 - K_o - e_{n+2}} \cdot \frac{K_o + e_{n+1}}{K_o + e_{n+2}} \cdot \frac{\Delta e_{n+1}}{\Delta e_n} \cdot \frac{1 - K_o - e_n}{1 - K_o - e_{n+1}}$$

$$\text{d'où } \lim_{n \rightarrow \infty} \frac{\Delta \epsilon_2^{(n+1)}}{\Delta \epsilon_2^{(n)}} = K_o \cdot \lim_{n \rightarrow \infty} \frac{\Delta e_{n+1}}{\Delta e_n}$$

Il nous faut donc à chaque étape des renseignements sur la suite à accélérer pour évaluer ce rapport. Cependant pour parler d'accélération il n'est pas nécessaire de connaître ce rapport mais simplement savoir s'il est différent de 1 et de 0.

Théorème 31 : soit  $\{S_n\}$  une suite convergente de limite  $S$ .

Si  $\lim_{n \rightarrow \infty} \left| \frac{\Delta S_{n+1}}{\Delta S_n} \right| = a_0 < 1$  alors :

$\lim_{n \rightarrow \infty} \left| \frac{\Delta \epsilon_2^{(n+1)}}{\Delta \epsilon_2^{(n)}} \right|$  existe et est égale à  $a_1 \Rightarrow a_1 \leq a_0 < 1$

Démonstration : elle est évidente à la suite du théorème précédent

si  $\lim_{n \rightarrow \infty} \left| \frac{\Delta \epsilon_{n+1}}{\Delta \epsilon_n} \right|$  existe elle est inférieure ou égale à 1

Ce théorème est intéressant pour l'étude du  $\Delta^2$  d'Aitken itéré en effet  $\lim_{n \rightarrow \infty} \left| \frac{\Delta S_{n+1}}{\Delta S_n} \right| < 1$  et si à chaque étape  $k$ ,  $\lim_{n \rightarrow \infty} \left| \frac{\Delta_k \epsilon_2^{(n+1)}}{\Delta_k \epsilon_2^{(n)}} \right|$  existe le  $\Delta^2$  d'Aitken accélère la convergence.

Nous allons nous intéresser maintenant aux autres colonnes du tableau  $\epsilon$  et en particulier le passage de  $\{\epsilon_{2k}^{(n)}\}$  à  $\{\epsilon_{2k+2}^{(n)}\}$

Théorème 32 : si  $\lim_{n \rightarrow \infty} \frac{\Delta \epsilon_{2k}^{(n+1)}}{\Delta \epsilon_{2k}^{(n)}} = a_k \neq 1$  et si  $\Delta \epsilon_{2k}^{(n)} = o(\Delta \epsilon_{2k-2}^{(n+1)})$

alors une condition nécessaire et suffisante pour que :

$\Delta \epsilon_{2k+2}^{(n)} = o(\Delta \epsilon_{2k}^{(n+1)})$  est que :

$$\lim_{n \rightarrow \infty} \Delta \epsilon_{2k+2}^{(n)} \Delta \epsilon_{2k+1}^{(n)} = 0$$

Démonstration : c'est l'application directe des théorèmes 18 et 19

Nous donnons maintenant le résultat de plus important de ce paragraphe en ce qui concerne la convergence et l'accélération de la convergence :

Théorème 33 : Si pour  $k$  fixé

-  $\epsilon_{2k}^{(n)}$  converge vers  $S$

-  $\lim_{n \rightarrow \infty} \left| \frac{\Delta \epsilon_{2k}^{(n+1)}}{\Delta \epsilon_{2k}^{(n)}} \right| = a_k \neq 0$  avec  $a_k < 1$

-  $\epsilon_{2k}^{(n)}$  converge plus vite que  $\epsilon_{2k-2}^{(n+1)}$

alors  $\epsilon_{2k+2}^{(n)}$  converge vers  $S$  plus vite que  $\epsilon_{2k}^{(n+1)}$  de plus si

$\lim_{n \rightarrow \infty} \left| \frac{\Delta \epsilon_{2k+2}^{(n+1)}}{\Delta \epsilon_{2k+2}^{(n)}} \right|$  existe et est égale à  $a_{k+1}$  alors  $a_{k+1} \leq a_k < 1$ .

Démonstration : il suffit d'appliquer les théorèmes 22 et 23 ainsi

que leur corollaire en effet si  $\epsilon_{2k}^{(n)}$  converge plus vite que  $\epsilon_{2k-2}^{(n+1)}$

alors  $A_k = 0$

Il nous reste donc à démontrer que  $a_{k+1} \leq a_k$  si  $a_{k+1}$  existe

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

$$\frac{\Delta \epsilon_{2k+2}^{(n+1)}}{\Delta \epsilon_{2k+2}^{(n)}} = \frac{\Delta \epsilon_{2k+1}^{(n+1)}}{\Delta \epsilon_{2k+1}^{(n+2)}} \left| \frac{\Delta \epsilon_{2k}^{(n+2)} \Delta \epsilon_{2k+1}^{(n+2)} + 1 - \frac{\Delta \epsilon_{2k+1}^{(n+2)}}{\Delta \epsilon_{2k+1}^{(n+1)}}}{\Delta \epsilon_{2k}^{(n+1)} \Delta \epsilon_{2k+1}^{(n+1)} + 1 - \frac{\Delta \epsilon_{2k+1}^{(n+1)}}{\Delta \epsilon_{2k+1}^{(n)}}} \right|$$

Si nous posons :

$$T_k^{(n)} = \Delta \epsilon_{2k}^{(n+1)} \Delta \epsilon_{2k}^{(n+1)} + 1 - \frac{\Delta \epsilon_{2k+1}^{(n+1)}}{\Delta \epsilon_{2k+1}^{(n)}}$$

nous avons  $\lim_{n \rightarrow \infty} T_k^{(n)} = 0$  si  $\epsilon_{2k+2}^{(n)}$  converge plus vite que  $\epsilon_{2k}^{(n+1)}$  de plus

si  $a_{k+1}$  existe alors  $\lim_{n \rightarrow \infty} \frac{T_k^{(n+1)}}{T_k^{(n)}}$  existe et puisque  $\lim_{n \rightarrow \infty} T_k^{(n)} = 0$

$$\lim_{n \rightarrow \infty} \left| \frac{T_k^{(n+1)}}{T_k^{(n)}} \right| \leq 1$$

donc  $a_{k+1} \leq a_k < 1$ .

Ce théorème est très intéressant en effet :

- si  $\{S_n\}$  est une suite convergente de limite  $S$

- si  $\lim_{n \rightarrow \infty} \left| \frac{\Delta S_{n+1}}{\Delta S_n} \right| = a_0 \neq 1$ , avec  $a_0 < 1$

- si  $\forall j = 1, 2, \dots, k$   $\lim_{n \rightarrow \infty} \left| \frac{\Delta \epsilon_{2j}^{(n+1)}}{\Delta \epsilon_{2j}^{(n)}} \right|$  existe alors à chaque

étape il y a accélération de la convergence vers  $S$ , de plus  $\{\epsilon_{2k+2}^{(n)}\}$  converge vers  $S$  plus vite que  $\{\epsilon_{2k}^{(n+1)}\}$ .

Nous pouvons également donner une nouvelle écriture de l' $\epsilon$ -algorithme.

$$\epsilon_{2k+2}^{(n)} = \frac{\sum_{i=0}^k \Delta \left( \frac{\epsilon^{(n+k-1)}}{\Delta \epsilon_{2i}^{(n+k-i)}} \right)}{\sum_{i=0}^k \Delta \left( \frac{1}{\Delta \epsilon_{2i}^{(n+k-i)}} \right)}$$

### III.2 LE $\rho$ -ALGORITHME.

Le  $\rho$ -algorithme consiste à faire passer une fraction rationnelle d'interpolation, dont numérateur et dénominateur sont des polynômes de degré  $k$ , par les  $2k+1$  couples de points  $(x_n, S_n), \dots, (x_{n+2k}, S_{n+2k})$  à l'aide d'une formule d'interpolation de Thiele (voir [40,4]) puis à calculer la valeur de cette fraction rationnelle lorsque  $n$  tend vers l'infini. Définissons d'abord ce que sont les différences réciproques d'une fonction. Soit une fonction dont on connaît la valeur  $S_n$  en un certain nombre de points  $x_n$ .

Définition 5 : on appelle différences réciproques les quantités

$$\rho_0^{(n)} = S_n \quad \forall n \in \mathbb{N}$$

$$\rho_1^{(n)} = \frac{x_n - x_{n+1}}{\rho_0^{(n)} - \rho_0^{(n+1)}}$$

$$\rho_2^{(n)} = \frac{x_n - x_{n+2}}{\rho_1^{(n)} - \rho_1^{(n+1)}} + \rho_0^{(n+1)}$$

---


$$\rho_k^{(n)} = \frac{x_n - x_{n+k}}{\rho_{k-1}^{(n)} - \rho_{k-1}^{(n+1)}} + \rho_{k-2}^{(n+1)}$$

On démontre que la fraction rationnelle  $R(x)$  dont numérateur et dénominateur sont des polynômes de degré  $k$  et telle que  $R(x_p) = S_p$  pour  $p=n, \dots, n+2k$  se met sous la forme

$$R(x) = \frac{\rho_{2k}^{(n)} x^k + \dots}{x^k + \dots}$$

Par conséquent nous avons  $\lim_{x \rightarrow \infty} \rho_{2k}^{(n)} = \rho_{2k}^{(n)}$  ce qui donne l'idée de prendre cette quantité  $\rho_{2k}^{(n)}$  comme approximation de la limite de la suite  $\{S_n\}$  lorsque  $n$  tend vers l'infini. Le calcul de  $\rho_{2k}^{(n)}$  s'effectue à l'aide de la forme étendue du  $\rho$ -algorithme qui n'est autre que le calcul des différences réciproques :

$$\rho_{-1}^{(n)} = 0 \quad \rho_0^{(n)} = S_n \quad \forall n \in \mathbb{N}$$

$$\rho_{k+1}^{(n)} = \rho_{k-1}^{(n+1)} + \frac{x_{n+k+1}^{-x_n}}{\rho_k^{(n+1)} - \rho_k^{(n)}}$$

On voit que la structure de cet algorithme est analogue à celle de l' $\epsilon$ -algorithme et que l'on peut construire un tableau identique au tableau  $\epsilon$ . Le  $\rho$ -algorithme a été utilisé pour la première fois comme transformation de suite à suite, afin d'accélérer la convergence, par Wynn [63] mais en se restreignant aux choix  $x_n = n \forall n \in \mathbb{N}$ . Dans certains cas il est préférable d'utiliser un autre choix pour les abscisses. Ainsi, si l'on applique le  $\rho$ -algorithme à la suite des approximations de la valeur d'une intégrale à l'aide de la formule des trapèzes avec des pas  $h_n = H/2^n$  il semble judicieux de prendre  $x_n = 1/h_n^2$ . On obtient ainsi une méthode proposée par Brezinski [14]. En utilisant un résultat de Gragg on montre que l'erreur est identique à celle faite par la méthode de Romberg mais avec un certain avantage pour le  $\rho$ -algorithme.

Sur l'extrapolation rationnelle on pourra consulter les références [16, 37, 54].

Le  $\rho$ -algorithme est peu utilisé en pratique sans doute à cause du manque de théorèmes de convergence le concernant. Nous allons ici énoncer certains résultats de convergence ; mais avant cela nous donne quelques propriétés :

Propriété 4 :

$$\rho_{2k}^{(n)} = \left| \begin{array}{cccccc} 1 & S_n & x_n & x_n \cdot S_n & \dots & x_n^{k-1} & x_n^{k-1} S_n & x_n^k S_n \\ \hline 1 & S_{n+2k} & x_{n+2k} & x_{n+2k} \cdot S_{n+2k} & \dots & \dots & \dots & x_{n+2k}^k S_{n+2k} \\ \hline 1 & S_n & x_n & x_n \cdot S_n & \dots & x_n^{k-1} & x_n^{k-1} S_n & x_n^k \\ \hline 1 & S_{n+2k} & x_{n+2k} & x_{n+2k} \cdot S_{n+2k} & \dots & \dots & \dots & x_{n+2k}^k \end{array} \right|$$

Théorème 34 : si on applique le  $\rho$ -algorithme à une suite  $\{S_n\}$  telle que :

$$S_n = \frac{S x_n^k + a_1 x_n^{k-1} + \dots + a_k}{x_n^k + b_1 x_n^{k-1} + \dots + b_k} \quad \text{alors}$$

$$\rho_{2k}^{(n)} = S \quad \forall n$$

Propriété 5 : si l'application du  $\rho$ -algorithme à  $\{S_n\}$  et à  $\{aS_n + b\}$  fournit respectivement les quantités  $\rho_k^{(n)}$  et  $\bar{\rho}_k^{(n)}$  alors :

$$\bar{\rho}_{2k}^{(n)} = a \rho_{2k}^{(n)} + b, \quad \bar{\rho}_{2k+1}^{(n)} = \rho_{2k+1}^{(n)} / a$$

Si l'application du  $\rho$ -algorithme à  $\{S_n\}$  et à  $\left\{\frac{aS_n + b}{cS_n + d}\right\}$  fournit respectivement les quantités  $\rho_k^{(n)}$  et  $\bar{\rho}_k^{(n)}$  alors :

$$\bar{\rho}_{2k}^{(n)} = \frac{a\rho_{2k}^{(n)} + b}{c\rho_{2k}^{(n)} + d}$$

Nous allons maintenant étudier le cas particulier de la deuxième colonne  $\rho_2^{(n)}$  que nous appelons le  $\rho_2$ -algorithme.

Théorème 35 : soit  $S_n$  une suite convergeant vers  $S$ . Si  $\lim_{n \rightarrow \infty} \Delta x_{n+1} / \Delta x_n$  existe et si

$$\lim_{n \rightarrow \infty} \frac{S_{n+1} - S}{S_n - S} = a_0 \neq \lim_{n \rightarrow \infty} \frac{\Delta x_{n+1}}{\Delta x_n} \text{ alors}$$

$\rho_2^{(n)}$  converge vers  $S$ .

Démonstration : évident d'après l'écriture de cette première étape.

Comme au chapitre précédent posons :

$$b_0^{(n)} = \frac{a_0^{(n+1)}}{a_0^{(n)}} = \frac{\Delta x_{n+1}}{\Delta x_n}$$

$$c_0^{(n)} = \frac{a_1^{(n)}}{a_0^{(n)}} = \frac{x_{n+2} - x_n}{x_{n+1} - x_n} = \frac{\Delta x_{n+1}}{\Delta x_n} + 1 = b_0^{(n)} + 1$$

et nous faisons les hypothèses

$$\lim_{n \rightarrow \infty} b_0^{(n)} = b_0$$

$$\lim_{n \rightarrow \infty} c_0^{(n)} = c_0$$

$$\lim_{n \rightarrow \infty} \frac{S_{n+1} - S}{S_n - S} = a_0$$

dans ces conditions nous pouvons dire :

Théorème 36 : si

$$- a_0 \neq 1, 0$$

$$- b_0 = \frac{1}{a_0}$$

$$- c_0 = \frac{1}{a_0} + 1$$

Alors  $\rho_2^{(n)} - S = o(S_{n+1} - S)$ .

Démonstration : nous appliquons les théorèmes 12 et 13 ainsi que leur corollaire : une condition nécessaire et suffisante d'accélération est

$$1 + b_0 = \frac{b_0 - a_0}{1 - a_0} \implies b_0 = \frac{1}{a_0}$$

d'où 
$$c_0 = 1 + \frac{1}{a_0}$$

C.Q.F.D.

Contrairement à l' $\epsilon$ -algorithme, il nous faut, pour le  $\rho$ -algorithme connaître  $a_0$ . Il peut venir à l'idée de prendre  $b_0^{(n)} = \frac{\Delta S_n}{\Delta S_{n+1}}$  les conditions du théorème sont remplies, dans ce cas  $\rho_2^{(n)} = \rho_2^{(n)}$  nous retrouvons donc l' $\epsilon$ -algorithme.

Là encore nous avons écarté le cas  $a_0 = 1$  cependant il est possible avec  $\rho$ -algorithme d'accélérer ce type de suite. Si  $\lim_{n \rightarrow \infty} \frac{S_{n+1} - S}{S_n - S} = a_0 = 1$  ce qui conduit à  $b_0 = 1$  et  $c_0 = 2$  qui est le  $\rho_2$ -algorithme simplifié de Wynn. Cependant ceci ne veut pas dire que le  $\rho_2$ -algorithme simplifié accélère toutes les suites telles que  $a_0 = 1$ . En effet supposons que :

$$\frac{S_{n+1} - S}{S_n - S} = 1 + e_n \text{ avec } \lim_{n \rightarrow \infty} e_n = 0$$

Théorème 37 : Si  $\lim_{n \rightarrow \infty} \frac{e_{n+1}}{e_n} \neq 1$  alors  $\lim_{n \rightarrow \infty} \rho_2^{(n)} = S$

Théorème 38 : soit  $\{S_n\}$  une suite convergeant vers  $S$ . Si

$$\lim_{n \rightarrow \infty} \left( \frac{1}{e_n} - \frac{\Delta x_{n+1}}{\Delta x_n} \frac{1}{e_{n+1}} \right) = 1 \text{ alors } \rho_2^{(n)} - S = o(S_{n+1} - S).$$

Démonstration : c'est une application directe du théorème 13 qui donne la condition suivante :

$$c_0 = 2 = 1 + \lim_{n \rightarrow \infty} \left( \frac{1}{e_n} - \frac{\Delta x_{n+1}}{\Delta x_n} \frac{1}{e_{n+1}} \right)$$

C.Q.F.D.

Ce théorème nous donne des renseignements sur le  $\rho_2$ -algorithme simplifié de Wynn ; en effet, cet algorithme accélère la convergence des suites à convergence logarithmique vérifiant les conditions :

$$\lim_{n \rightarrow \infty} \frac{e_{n+1}}{e_n} \neq 1$$

$$\lim_{n \rightarrow \infty} \Delta\left(\frac{1}{e_n}\right) = -1$$

Maintenant nous nous intéressons au cas  $\lim_{n \rightarrow \infty} \frac{e_{n+1}}{e_n} = 1$  où encore

$$\lim_{n \rightarrow \infty} \frac{\Delta S_{n+1}}{\Delta S_n} = 1 \text{ ce que nous écrirons sous la forme :}$$

$$\frac{\Delta S_{n+1}}{\Delta S_n} = 1 + d_n \text{ avec } \lim_{n \rightarrow \infty} d_n = 0.$$

Pour donner un résultat d'accélération de la convergence nous devons nous placer dans les hypothèses suivantes :

$$b_0^{(n)} = b_0 = 1 \text{ et } c_0^{(n)} = c_0 = 2$$

ce qui est le cas du  $\rho$ -algorithme simplifié de Wynn.

Théorème 39 : dans les hypothèses ci-dessus. Si :

$$\lim_{n \rightarrow \infty} \frac{1}{1 + \frac{\Delta(\frac{1}{d_n})}{d_n}} = 2 \text{ alors}$$

$$\Delta \rho_2^{(n)} = o(\Delta S_{n+1})$$

Démonstration : c'est une application directe du théorème 16.

Nous nous intéressons maintenant aux autres colonnes du tableau  $\rho$ .

Nous posons pour cela : en reprenant la notation du chapitre II

$$b_k^{(n)} = \frac{a_{2k}^{(n+1)}}{a_{2k}^{(n)}}$$

$$c_k^{(n)} = \frac{a_{2k+1}^{(n)}}{a_{2k}^{(n)}}$$

et nous nous plaçons dans les hypothèses suivantes :

$$\lim_{n \rightarrow \infty} \frac{\Delta \rho_{2k}^{(n+1)}}{\Delta \rho_{2k}^{(n)}} = a_k \quad \forall k = 0, 1, \dots$$

$$\lim_{n \rightarrow \infty} b_k^{(n)} = b_k$$

$$\lim_{n \rightarrow \infty} c_k^{(n)} = c_k$$

Considérons le passage de la colonne  $2k$  à la colonne  $2k+2$ .

Théorème 40 : si

-  $\{\rho_{2k}^{(n)}\}$  converge vers S plus vite que  $\{\rho_{2k-2}^{(n+1)}\}$

$$- \lim_{n \rightarrow \infty} \left| \frac{\Delta \rho_{2k}^{(n+1)}}{\Delta \rho_{2k}^{(n)}} \right| = |a_k| < 1$$

$$- b_k = 1 + \frac{a_0^{2k}(1-a_0)}{1-a_0^{2k+1}} \left[ \frac{1}{a_k} - 1 \right]$$

$$- c_k = 1 + \frac{1}{a_k} \left[ \frac{a_0^{2k}(1-a_0)}{1-a_0^{2k+1}} \right]$$

alors  $\{\rho_{2k+2}^{(n)}\}$  converge vers S plus vite que  $\{\rho_{2k}^{(n+1)}\}$  de plus si

$$\lim_{n \rightarrow \infty} \left| \frac{\Delta \rho_{2k+2}^{(n+1)}}{\Delta \rho_{2k+2}^{(n)}} \right| \text{ existe alors } |a_{k+1}| \leq |a_k| < 1.$$

Démonstration :

$$b_k^{(n)} = \frac{a_{2k}^{(n+1)}}{a_{2k}^{(n)}} = \frac{x_{n+2k+2} - x_{n+1}}{x_{n+2k+1} - x_n}$$

$$b_k^{(n)} = \frac{x_{n+2k+1}}{x_{n+2k+1} - x_n} + 1 - \frac{\Delta x_n}{x_{n+2k+1} - x_n}$$

$$c_k^{(n)} = \frac{a_{2k+1}^{(n)}}{a_{2k}^{(n)}} = \frac{x_{n+2k+2} - x_n}{x_{n+2k+1} - x_n} = \frac{\Delta x_{n+2k+1}}{x_{n+2k+1} - x_n} + 1$$

$$\text{d'où } c_k^{(n)} = b_k^{(n)} + \frac{\Delta x_n}{x_{n+2k+1} - x_n}$$

si nous appliquons le théorème 14 et son corollaire :

$$c_k = b_k + \lim_{n \rightarrow \infty} \frac{\Delta x_n}{x_{n+2k+1} - x_n} = \frac{a_k - b_k}{a_k - 1}$$

$$\text{notons } T = \lim_{n \rightarrow \infty} \frac{\Delta x_n}{x_{n+2k+1} - x_n}$$

$$\Rightarrow b_k = 1 + T \left[ \frac{1}{a_k} - 1 \right]$$

$$c_k = 1 + \frac{T}{a_k}$$

$$T = \lim_{n \rightarrow \infty} \frac{1}{\frac{2k \Delta x_{n+i}}{\sum_{i=0}^{2k} \Delta x_n}} = \frac{1}{1 + \sum_{i=1}^{2k} b_0^i}$$

et nous avons  $b_0 = \frac{1}{a_0}$  pour assurer l'accélération du  $\rho_2$ -algorithme.

C.Q.F.D.

Ce théorème nous donne la condition que doit vérifier  $b_k$  et  $c_k$  à chaque étape pour qu'il y ait accélération de la convergence. Cependant ceci est difficilement vérifiable car il faut connaître les  $a_k$ .

Remarque. Si  $a_j = 1 \forall j \in \mathbb{N}$  le  $\rho$ -algorithme simplifié vérifie les conditions d'accélération mais ceci n'est qu'une remarque et non une démonstration car nous appliquons un théorème donc l'une des hypothèses est  $a_j \neq 1$ .

Enfin nous pouvons donner une nouvelle écriture du  $\rho$ -algorithme

$$\binom{(n)}{2k+2} = \frac{\sum_{i=0}^k \Delta \left[ \frac{-\binom{(n+k-i)}{2i} \rho_{2i}^{(n+k-i)}}{\Delta \rho_{2i}^{(n+k-i)}} \right] + \sum_{i=0}^k (a_{2i+1}^{n+k-i} - a_{2i}^{(n+k-i)})}{\sum_{i=0}^k \Delta \left[ \frac{a_{2i}^{(n+k-1)}}{\Delta \rho_{2i}^{(n+k-i)}} \right]}$$

où  $a_k^{(n)} = x_{n+k+1} - x_n \quad \forall k \text{ et } n \in \mathbb{N}$

### III,3 GÉNÉRALISATIONS DE L' $\epsilon$ -ALGORITHME

La seule différence entre les règles de l' $\epsilon$ -algorithme et du  $\rho$ -algorithme est l'introduction d'une suite de paramètres  $\{x_n\}$ . Il est donc tentant d'essayer d'introduire également des paramètres dans l' $\epsilon$ -algorithme. Ceci a été effectué de deux façons qui aboutissent à deux généralisations de l' $\epsilon$ -algorithme :

la première généralisation est la suivante :

$$\begin{aligned} \epsilon_{-1}^{(n)} &= 0 & \epsilon_0^{(n)} &= S_n & \text{pour } n \in \mathbb{N} \\ \epsilon_{k+1}^{(n)} &= \epsilon_{k-1}^{(n+1)} + \frac{\Delta x_n}{\epsilon_k^{(n+1)} - \epsilon_k^{(n)}} & n, k &\in \mathbb{N} \end{aligned}$$

Nous allons donner quelques résultats théoriques : nous définissons d'abord l'opérateur  $R$  : soit  $f$  une fonction des  $n$  quantités  $S_{p_1}, \dots, S_{p_n}$  et posons  $r_n = \Delta S_n / \Delta x_n$  alors :

$$R f(S_{p_1}, \dots, S_{p_n}) = f(r_{p_1}, \dots, r_{p_n})$$

$R$  est une généralisation de l'opérateur  $\Delta$  que l'on retrouve si  $\Delta x_n = 1$ .

Théorème 41 : on a

$$\epsilon_{2k}^{(n)} = \begin{array}{|c} v_n \dots\dots\dots v_{n+k} \\ Rv_n \dots\dots\dots Rv_{n+k} \\ \hline R^k v_n \dots\dots\dots R^k v_{n+k} \\ \hline \Delta x_n \dots\dots\dots \Delta x_{n+k} \\ Rv_n \dots\dots\dots Rv_{n+k} \\ R^k v_n \dots\dots\dots R^k v_{n+k} \end{array}$$

et

$$\epsilon_{2k+1}^{(n)} = \begin{array}{|c} \Delta x_n \dots\dots\dots \Delta x_{n+k} \\ R^2 v_n \dots\dots\dots R^2 v_{n+k} \\ \hline R^{k+1} v_n \dots\dots\dots R^{k+1} v_{n+k} \\ \hline Rv_n \dots\dots\dots Rv_{n+k} \\ \hline R^{k+1} v_n \dots\dots\dots R^{k+1} v_{n+k} \end{array}$$

Nous pouvons tirer de ce résultat la forme des suites  $\{S_n\}$  pour lesquelles cette première généralisation donne le résultat exact.

Théorème 42 : une condition nécessaire et suffisante pour que  $\epsilon_{2k}^{(n)} = S$   
 $\forall n > N$  est qu'il existe  $a_0, \dots, a_k$  non tous nuls tels que :

$$a_0(v_n - S \cdot x_n) + \sum_{i=1}^k a_i \cdot R^i v_n = 0 \quad \forall n > N.$$

Nous pouvons tirer du second chapitre les deux résultats suivants :

Propriété 6 : si l'application de la première généralisation de l' $\epsilon$ -algorithme à  $\{S_n\}$  et  $x \{aS_n + b\}$  fournit respectivement les quantités  $\epsilon_k^{(n)}$  et  $\bar{\epsilon}_k^{(n)}$  alors

$$\bar{\epsilon}_{2k}^{(n)} = a \epsilon_{2k}^{(n)} + b \quad \bar{\epsilon}_{2k+1}^{(n)} = \epsilon_{2k+1}^{(n)} / a$$

Théorème 43 : une condition nécessaire et suffisante pour que

$\epsilon_2^{(n)} = S \forall n > N$  est que :

$$S_n = S + A \prod_{i=0}^{n-1} \lambda_i \text{ avec } \lambda_i = 1 + c \cdot \Delta x_i$$

Démonstration : dans le théorème 10  $a_0^{(i)} = a_1^{(i)} = \Delta x_i$   
 $\rho_0 = c \cdot \Delta x_0$  et  $\rho_i = c \cdot \Delta x_i$  ou  $c = \frac{1}{B}$  et  $S_n = S + A \cdot \prod_{i=0}^{n-1} (1 + c \cdot \Delta x_i)$

Nous allons étudier maintenant la convergence en particulier nous faisons d'abord l'étude de  $\{\epsilon_2^{(n)}\}$ .

Notons :

$$b_k^{(n)} = \frac{\Delta x_{n+1}}{\Delta x_n} \quad \forall n, k \in \mathbb{N}$$

$$c_k^{(n)} = 1 \quad \forall n, k \in \mathbb{N}$$

et nous faisons l'hypothèse suivante :

$$\lim_{n \rightarrow \infty} b_k^{(n)} = b_k$$

Théorème 44 : soit  $S_n$  une suite convergeant vers  $S$  si :

$$- \lim_{n \rightarrow \infty} \frac{S_{n+1} - S}{S_n - S} = a_0 \neq 1, 0$$

$$- b_0 \neq a_0$$

$$\text{alors } \lim_{n \rightarrow \infty} \epsilon_2^{(n)} = S$$

si de plus  $b_0 = 1$  alors :

$$\epsilon_2^{(n)} - S = o(S_{n+1} - S)$$

Démonstration : c'est une application directe des théorèmes 12 et 13 en

$$\text{effet } A_0 = 0 \quad b_0 \neq a_0 \implies \lim_{n \rightarrow \infty} S_n = \lim_{n \rightarrow \infty} \epsilon_2^{(n)} = S$$

$$\text{et } c_0 = \frac{a_0 - b_0}{a_0 - 1} = 1 \iff b_0 = 1 \iff (\epsilon_2^{(n)} - S) = o(S_{n+1} - S)$$

C.Q.F.D.

Nous pouvons supposer un procédé itéré de cette première généralisation de l' $\epsilon$ -algorithme et à chaque étape changer la suite auxiliaire  $x_n$ . Il est alors nécessaire de savoir si :  $\lim_{n \rightarrow \infty} \left| \frac{\Delta \epsilon_2^{(n+1)}}{\Delta \epsilon_2^{(n)}} \right|$  existe et si cette limite est strictement inférieure à un.

Théorème 45 : soit  $\{S_n\}$  une suite convergente vers  $S$ . Si :

$$- \lim_{n \rightarrow \infty} \frac{\Delta S_{n+1}}{\Delta S_n} = a_0 \neq 0 \text{ avec } |a_0| < 1$$

$$- b_0 = 1$$

$$\text{alors : si } \lim_{n \rightarrow \infty} \frac{\Delta \epsilon_2^{(n+1)}}{\Delta \epsilon_2^{(n)}} = a_1 \text{ nous avons } |a_1| \leq |a_0| < 1.$$

Démonstration : si  $b_0 = 1$  et  $|a_0| < 1$  alors  $\{\epsilon_2^{(n)}\}$  converge vers  $S$  plus vite que  $S_{n+1}$ . Dans ce cas nous pouvons mettre

$$\frac{\Delta\epsilon_2^{(n+1)}}{\Delta\epsilon_2^{(n)}} \text{ sous la forme } \frac{\Delta\epsilon_1^{(n+1)}}{\Delta\epsilon_1^{(n+2)}} \cdot \frac{T_n^{(n+1)}}{T_0^{(n)}} \text{ où } \lim_{n \rightarrow \infty} T_0^{(n)} = 0 \text{ dans ce cas si}$$

$$\lim_{n \rightarrow \infty} \frac{\Delta\epsilon_2^{(n+1)}}{\Delta\epsilon_2^{(n)}} \text{ existe et est égale à } a_1 \text{ alors } |a_1| = |a_0| \cdot \lim_n \left| \frac{T_0^{(n+1)}}{T_0^{(n)}} \right| < |a_0|$$

C.Q.F.D.

Ce théorème nous donne une condition pour que l'application itérée de la première généralisation de l' $\epsilon$ -algorithme accélère la convergence à chaque étape. En effet si  $|a_0| < 1$ ,  $b_0 = 1$  et si nous notons  $\epsilon_2^{(n)}$  : l'application itérée  $k$  fois de cet algorithme, une condition pour qu'il y ait accélération à l'étape  $k$  est que :  $\forall j=1, \dots, k-1$

$$\lim_{n \rightarrow \infty} \frac{\Delta_j \epsilon_2^{(n+1)}}{\Delta_j \epsilon_2^{(n)}} \text{ existe. Nous pouvons étudier la suite } \Delta x_n \text{ pour que cette con-}$$

dition soit vérifiée.

Nous avons écarté, dans les théorèmes ci-dessus, le cas d'une suite à convergence logarithmique, c'est-à-dire lorsque  $a_0 = 1$ . Cependant il est possible d'accélérer la convergence de ce type de suite. Pour cela nous faisons l'hypothèse suivante :  $\frac{S_{n+1} - S}{S_n - S} = 1 + e_n$  avec  $\lim_{n \rightarrow \infty} e_n = 0$ .

Théorème 46 : soit  $\{S_n\}$  une suite de limite  $S$  vérifiant l'hypothèse précédente, alors

$$\left. \begin{aligned} b_0 &= \lim_{n \rightarrow \infty} \frac{e_{n+1}}{e_n} \\ \lim_{n \rightarrow \infty} \left( \frac{1}{e_n} - \frac{\Delta x_{n+1}}{\Delta x_n} \frac{1}{e_{n+1}} \right) &= 0 \end{aligned} \right\} \Leftrightarrow \epsilon_2^{(n)} = S + o(S_{n+1} - S)$$

Démonstration : ce théorème est une application directe de théorème 15.

Nous nous intéressons maintenant aux autres colonnes du tableau  $\epsilon$ . En particulier nous étudions le passage de la colonnes  $2k$  à la colonnes  $2k+2$ .

Théorème 47 : soit une suite  $\{S_n\}$  qui converge vers  $S$ . Si  $\{\epsilon_{2k}^{(n)}\}$  converge vers  $S$  plus vite que  $\{\epsilon_{2k-2}^{(n+2)}\}$  alors

$$b_k = b_0 \neq a_k \implies \lim_{n \rightarrow \infty} \epsilon_{2k+2}^{(n)} = S$$

Démonstration : c'est une application directe du théorème 22 et son corollaire.

Théorème 48 : soit  $\{S_n\}$  une suite de limite  $S$ . Si

$$- \lim_{n \rightarrow \infty} \left| \frac{\Delta \epsilon_{2k}^{(n+1)}}{\Delta \epsilon_{2k}^{(n)}} \right| = |a_k| < 1$$

$$- \{\epsilon_{2k}^{(n)}\} \text{ converge vers } S \text{ plus vite que } \{\epsilon_{2k-2}^{(n+1)}\}$$

$$- b_k = b_0 = 1$$

$$\text{alors } (\epsilon_{2k+2}^{(n)} - S) = o(\epsilon_{2k}^{(n+1)} - S)$$

$$\text{de plus si } \lim_{n \rightarrow \infty} \left| \frac{\Delta \epsilon_{2k+2}^{(n+1)}}{\Delta \epsilon_{2k+2}^{(n)}} \right| \text{ existe alors } |a_{k+1}| \leq |a_k| < 1.$$

Démonstration : en effet si  $|a_k| < 1$  et  $(\epsilon_{2k}^{(n)} - S) = o(\epsilon_{2k-2}^{(n+1)} - S)$  alors  $A_k = 0$  dans le théorème 23 or  $a_k \neq b_k \implies \lim_{n \rightarrow \infty} \epsilon_{2k+2}^{(n)} = S$  et  $b_k = 1 \implies c_k = 1 \implies \epsilon_{2k+2}^{(n)} - S = o(\epsilon_{2k}^{(n+1)} - S)$  de plus dans ce cas nous pouvons mettre

$$\frac{\Delta \epsilon_{2k+2}^{(n+1)}}{\Delta \epsilon_{2k}^{(n)}} \text{ sous la forme } \frac{\Delta \epsilon_{2k+1}^{(n+1)}}{\Delta \epsilon_{2k}^{(n+2)}} \cdot \frac{T_k^{(n+1)}}{T_k^{(n)}} \text{ avec } \lim_{n \rightarrow \infty} T_k^{(n)} = 0$$

alors si  $\lim_{n \rightarrow \infty} \frac{\Delta \epsilon_{2k+2}^{(n+1)}}{\Delta \epsilon_{2k+2}^{(n)}}$  existe alors

$$|a_{k+1}| < |a_k| \cdot \lim_{n \rightarrow \infty} \left| \frac{T_k^{(n+1)}}{T_k^{(n)}} \right| \leq |a_k| < 1.$$

C.Q.F.D.

Ce théorème est important en effet si  $|a_0| < 1$  et si  $\forall j \lim_{n \rightarrow \infty} \left| \frac{\Delta \epsilon_{2j}^{(n+1)}}{\Delta \epsilon_{2j}^{(n)}} \right|$

existe alors la première généralisation de l' $\epsilon$ -algorithme accélère la convergence de la colonne précédente à chaque étape. Il est sans doute possible d'étudier la suite  $x_n$  pour qu'il y ait existence de cette limite.

Nous pouvons enfin mettre cette première généralisation de l' $\epsilon$ -algorithme sous la forme :

$$\epsilon_{2k+2}^{(n)} = \frac{\sum_{i=0}^k \Delta \left[ \frac{\Delta x_{n+k-i} \cdot \epsilon_{2i}^{(n+k-i)}}{\Delta \epsilon_{2i}^{(n+k-i)}} \right]}{\sum_{i=0}^k \Delta \left[ \frac{\Delta x_{n+k-i}}{\Delta \epsilon_{2i}^{(n+k-i)}} \right]}$$

La seconde généralisation de l' $\epsilon$ -algorithme est donnée par :

$$\epsilon_{-1}^{(n)} = 0 \quad \epsilon_0^{(n)} = S_n \quad \forall n \in \mathbb{N}$$

$$\epsilon_{k+1}^{(n)} = \epsilon_{k-1}^{(n+1)} + \frac{\Delta x_{n+k}}{\epsilon_k^{(n+1)} - \epsilon_k^{(n)}} \quad \forall n, k \in \mathbb{N}$$

Il n'a pas été possible d'obtenir, pour cette seconde généralisation des résultats analogues aux théorèmes 41, 42. Cependant nous pouvons énoncer certains résultats :

Propriété 2 : si l'application de la seconde généralisation de l' $\epsilon$ -algorithme à  $\{S_n\}$  et à  $\{aS_n + b\}$  fournit respectivement les quantités  $\epsilon_k^{(n)}$  et  $\bar{\epsilon}_k^{(n)}$  alors :

$$\bar{\epsilon}_{2k}^{(n)} = a \epsilon_{2k}^{(n)} + b \quad \bar{\epsilon}_{2k+1}^{(n)} = \epsilon_{2k+1}^{(n)} / a$$

Démonstration : il suffit d'appliquer la propriété 1.

Nous pouvons également donner les suites pour lesquelles la première étape de la seconde généralisation de l' $\epsilon$ -algorithme fournit le résultat exact :

Théorème 49 : une condition nécessaire et suffisante pour que  $\epsilon_2^{(n)} = S$   $\forall n > N$  est que :

$$S_n = S + A \prod_{i=0}^{n-1} \lambda_i \quad \forall n > N \text{ avec } \lambda_i = \frac{1}{1 + C \cdot \Delta x_i}$$

Démonstration : d'après le théorème 10 nous obtenons les suites de la forme :

$$S_n = S + A \prod_{i=0}^{n-1} (1 + \rho_i) \text{ avec } \rho_0 = \frac{a_0^{(0)}}{B} \text{ et } \rho_i = \frac{a_0^{(i)}}{B + \sum_{j=0}^{i-1} (a_0^{(j)} - a_1^{(j)})}$$

dans le cas présent nous avons :

$$\rho_0 = \frac{\Delta x_0}{B} \text{ si nous posons } c = - \frac{1}{\Delta x_0 + E}$$

$$1 + \rho_0 = \frac{1}{1+c.\Delta x_0}$$

$$1 + \rho_i = 1 + \frac{x_i}{i-1} = 1 + \frac{x_i}{B + \sum_{j=0}^{i-1} (\Delta x_j - \Delta x_{j+1})} = 1 + \frac{x_i}{B + \Delta x_0 - x_i}$$

$$1 + \rho_i = \frac{1}{1+c.\Delta x_i} \quad \text{d'où le résultat}$$

Nous étudions maintenant la convergence de cet algorithme ; pour cela nous posons :

$$b_k^{(n)} = \frac{\Delta x_{2k+n+1}}{\Delta x_{n+2k}} = c_k^{(n)}$$

et nous faisons l'hypothèse :

$$\lim_{n \rightarrow \infty} b_k^{(n)} = \lim_{n \rightarrow \infty} c_k^{(n)} = b_k$$

Nous étudions d'abord la première étape de l'algorithme soit  $\{\epsilon_2^{(n)}\}$  :

Théorème 50 : soit  $S_n$  une suite de limite  $S$  si :

$$- \lim_{n \rightarrow \infty} \frac{\Delta S_{n+1}}{\Delta S_n} = a_0 \neq 1, 0$$

$$- b_0 = a_0$$

alors  $\lim_{n \rightarrow \infty} \epsilon_2^{(n)} \neq S$

si de plus  $b_0 = 1$  alors :

$$\epsilon_2^{(n)} - S = o(S_{n+1} - S)$$

Démonstration : c'est une application des théorèmes 12 et 13 en effet  $A_0 = 0$

$$b_0 \neq a_0 \implies \lim_{n \rightarrow \infty} \varepsilon_2^{(n)} = \lim_{n \rightarrow \infty} S_n = S \text{ et } c_0 = b_0 = \frac{a_0 - b_0}{a_0 - 1} \iff b_0 = c_0 = 1 \iff$$

$$(\varepsilon_2^{(n)} - S) = o(S_{n+1} - S)$$

C.Q.F.D.

Nous pouvons imaginer une application itérée de cette seconde généralisation de l' $\varepsilon$ -algorithme et à chaque étape changer la suite auxiliaire  $x_n$ . Il nous est alors nécessaire de savoir si  $\lim_{n \rightarrow \infty} \left| \frac{\Delta \varepsilon_2^{(n+1)}}{\Delta \varepsilon_2^{(n)}} \right|$  existe et si cette limite est strictement inférieure à un :

Théorème 51 : soit  $\{S_n\}$  une suite de limite  $S$ . Si :

$$- \lim_{n \rightarrow \infty} \frac{\Delta S_{n+1}}{\Delta S_n} = a_0 \neq 0 \text{ avec } |a_0| < 1$$

$$- b_0 = 1$$

$$\text{alors si } \lim_{n \rightarrow \infty} \frac{\Delta \varepsilon_2^{(n+1)}}{\Delta \varepsilon_2^{(n)}} = a_1 \text{ nous avons } |a_1| \leq |a_0| < 1.$$

Démonstration : si  $b_0 = 1$  et  $|a_1| < 1$  alors  $\{\varepsilon_2^{(n)}\}$  converge vers  $S$  plus vite que  $\{S_{n+1}\}$ . Dans ce cas nous pouvons mettre le rapport

$$\frac{\Delta \varepsilon_2^{(n+1)}}{\Delta \varepsilon_2^{(n)}} \text{ sous la forme } \frac{\Delta \varepsilon_1^{(n+1)}}{\Delta \varepsilon_1^{(n+2)}} \cdot \frac{T_0^{(n+1)}}{T_0^{(n)}} \text{ où } \lim_{n \rightarrow \infty} T_0^{(n)} = 0 \text{ dans ce cas si}$$

$$\lim_{n \rightarrow \infty} \frac{\Delta \varepsilon_2^{(n+1)}}{\Delta \varepsilon_2^{(n)}} \text{ existe alors } |a_1| = |a_0| \lim_{n \rightarrow \infty} \left| \frac{T_0^{(n+1)}}{T_0^{(n)}} \right| \leq |a_0|$$

C.Q.F.D.

Donc si nous notons  $\epsilon_2^{(n)}$  la première étape de l'algorithme itérée k fois, pour avoir accélération de la convergence il suffit que :

$$- |a_0| < 1$$

$$- \text{et de construire } x_n \text{ pour que les limites de } \frac{\Delta_j \epsilon_2^{(n+1)}}{\Delta_j \epsilon_2^{(n)}} \text{ existent}$$

quel que soit j.

Là également nous avons écarté le cas  $a_0 = 1$ , il est cependant possible d'accélérer certaines de ces suites par k deuxième généralisation de l' $\epsilon$ -algorithme. Posons alors  $\frac{S_{n+1} - S}{S_n - S} = 1 + e_n$  avec  $\lim_{n \rightarrow \infty} e_n = 0$ .

Théorème 52 : soit  $\{S_n\}$  une suite de limite S et vérifiant l'hypothèse ci-dessus, alors :

$$\left. \begin{aligned} b_0 &= \lim_{n \rightarrow \infty} \frac{e_{n+1}}{e_n} \\ \lim_{n \rightarrow \infty} \frac{\Delta x_{n+1}}{\Delta x_n} &= 1 + \lim_{n \rightarrow \infty} \left( \frac{1}{e_n} - \frac{\Delta x_{n+1}}{\Delta x_n} \frac{1}{e_{n+1}} \right) \end{aligned} \right\} \iff \epsilon_2^{(n)} - S = 0 \quad (S_{n+1} - S)$$

Démonstration : ce théorème est une application directe du théorème 15.

Nous pouvons remarquer que si  $\lim_{n \rightarrow \infty} \frac{e_{n+1}}{e_n} = 1$  la condition devient

$$b_0 = \lim_{n \rightarrow \infty} \frac{e_{n+1}}{e_n} \text{ et } \lim_{n \rightarrow \infty} \left( \frac{1}{e_n} - \frac{\Delta x_{n+1}}{\Delta x_n} \frac{1}{e_{n+1}} \right) = 0.$$

Nous nous intéressons maintenant aux autres colonnes du tableau .

En particulier le passage de  $\{\epsilon_{2k}^{(n)}\}$  à  $\{\epsilon_{2k+2}^{(n)}\}$ .

Théorème 53 : soit une suite  $\{S_n\}$  de limite  $S$ . Si  $\{\epsilon_{2k}^{(n)}\}$  converge vers  $S$  plus vite que  $\{\epsilon_{2k-2}^{(n+1)}\}$  alors :

$$b_k = b_0 \neq a_k \implies \lim_{n \rightarrow \infty} \epsilon_{2k+2}^{(n)} = S$$

La démonstration est analogue à celle du théorème 47.

Théorème 54 : soit  $\{S_n\}$  une suite de limite  $S$ . Si

$$\begin{aligned} - \lim_{n \rightarrow \infty} \left| \frac{\Delta \epsilon_{2k}^{(n+1)}}{\Delta \epsilon_{2k}^{(n)}} \right| &= |a_k| < 1 \\ - \{\epsilon_{2k}^{(n)}\} &\text{ converge vers } S \text{ plus vite que } \{\epsilon_{2k-2}^{(n+1)}\} \\ - b_k &= b_0 = 1 \end{aligned}$$

alors  $(\epsilon_{2k+2}^{(n)} - S) = o(S_{n+1} - S)$

de plus si  $\lim_{n \rightarrow \infty} \left| \frac{\Delta \epsilon_{2k+2}^{(n+1)}}{\Delta \epsilon_{2k+2}^{(n)}} \right|$  existe et est égale à  $|a_{k+1}|$  alors  $|a_{k+1}| \leq |a_k| < 1$

démonstration analogue à celle du théorème 48.

Là encore la seule condition restreignante est l'existence des limites de  $\frac{\Delta \epsilon_{2j}^{(n+1)}}{\Delta \epsilon_{2j}^{(n)}}$

Nous pouvons enfin mettre cette seconde généralisation de l' $\epsilon$ -algorithme sous la forme

$$\epsilon_{2k+2}^{(n)} = \frac{\sum_{i=0}^k \Delta \left( \frac{\Delta x_{n+k+i} \epsilon_{2i}^{(n+k-i)}}{\Delta \epsilon_{2i}^{(n+k-i)}} \right)}{\sum_{i=0}^k \Delta \left( \frac{\Delta x_{n+k+i}}{\Delta \epsilon_{2i}^{(n+k-i)}} \right)}$$

Voici deux exemples qui montrent l'efficacité de ces algorithmes dans certains cas.

Exemple 1 :

$S_n = 1 + 1/n$  suite à convergence logarithmique

$x_n = \text{Log}(1+n)$  ; cette suite auxiliaire vérifie les conditions des théorèmes 46 et 52

on obtient :

$S_{37} = 1,027$   $\epsilon_{36}^{(0)} = 1,004$  avec l' $\epsilon$ -algorithme

$\epsilon_{36}^{(0)} = 1,000008$  avec la première généralisation

$\epsilon_{36}^{(0)} = 1,00000002$  avec la seconde généralisation.

Exemple 2 :

$S_n = n$  sur  $\frac{1}{n}$  et  $x_n = \log(1+n)$

$S_{37} = 0,99987$   $\epsilon_{36}^{(0)} = 0,9999938$  avec l' $\epsilon$ -algorithme

$\epsilon_{36}^{(0)} = 1,000000027$  avec la première généralisation

$\epsilon_{36}^{(0)} = 0,9999999976$  avec la seconde généralisation

## CHAPITRE IV

### ETUDE DU PROCÉDÉ Q

Le procédé q est un procédé d'accélération de la convergence qui a été énoncé et étudié par BREZINSKI [3,66]. Les transformés obtenus s'écrivent sous la forme d'un rapport de deux déterminants ce qui rend leur évaluation difficile. Nous donnons ici un algorithme récursif permettant le calcul de ce rapport ; cet algorithme est calqué sur l'algorithme  $G_n$  [36, 44, 46]. Cette nouvelle formulation nous permet une étude de la convergence et de l'accélération de convergence du procédé q. Enfin nous appliquons les résultats obtenus au calcul d'intégrales impropres et à l' $\epsilon$ -algorithme.

#### IV.1 NOTATIONS ET PRÉLIMINAIRES.

Nous pouvons considérer le procédé q comme une modification de la transformation de Shanks. Il peut être défini par le rapport de deux déterminants :

$$q_{k+1}(x_n, S_n) = \frac{\begin{vmatrix} S_n & x_n & \dots & x_{n+k} \\ S_{n+1} & x_{n+1} & \dots & x_{n+k+1} \\ \dots & \dots & \dots & \dots \\ S_{n+k+1} & x_{n+k+1} & \dots & x_{n+2k+1} \end{vmatrix}}{\begin{vmatrix} \Delta x_n & \Delta x_{n+1} & \dots & \Delta x_{n+k} \\ \Delta x_{n+1} & \Delta x_{n+2} & \dots & \Delta x_{n+k+1} \\ \dots & \dots & \dots & \dots \\ \Delta x_{n+k} & \Delta x_{n+k+1} & \dots & \Delta x_{n+2k} \end{vmatrix}}$$

où  $S_n$  est une suite de nombres réels et  $x_n$  une suite auxiliaire.

Pour calculer ce rapport de deux déterminants nous allons le transformer.

Pour cela notons :

$$G_k(x_n, S_n) = \frac{\begin{vmatrix} S_n & \dots & S_{n+k} \\ x_n & \dots & x_{n+k} \\ \hline x_{n+k-1} & \dots & x_{n+2k-1} \end{vmatrix}}{\begin{vmatrix} 1 & 1 & \dots & 1 \\ x_n & x_{n+1} & \dots & x_{n+k} \\ \hline x_n & x_{n+k} & \dots & x_{n+2k-1} \end{vmatrix}}$$

pour  $k \geq 1$  et  $G_0(x_n, S_n) = S_n$ .

Maintenant retranchant à chaque colonne la précédente dans le déterminant du dénominateur :

$$G_k(x_n, S_n) = \frac{\begin{vmatrix} S_n & S_{n+1} & \dots & S_{n+k} \\ x_n & x_{n+1} & \dots & x_{n+k} \\ \hline x_{n+k-1} & x_{n+k} & \dots & x_{n+2k-1} \end{vmatrix}}{\begin{vmatrix} 1 & 1 & \dots & 0 \\ x_n & \Delta x_n & \dots & \Delta x_{n+k-1} \\ \hline x_{n+k-1} & \Delta x_{n+k-1} & \dots & x_{n+2k-2} \end{vmatrix}}$$

qui donne :

$$G_k(x_n, S_n) = \frac{\begin{vmatrix} S_n & S_{n+1} & \dots & S_{n+k} \\ x_n & x_{n+1} & \dots & x_{n+k} \\ \dots & \dots & \dots & \dots \\ x_{n+k-1} & x_{n+k} & \dots & x_{n+2k-1} \end{vmatrix}}{\begin{vmatrix} \Delta x_n & \Delta x_{n+1} & \dots & \Delta x_{n+k-1} \\ \Delta x_{n+1} & \Delta x_{n+2} & \dots & \Delta x_{n+k} \\ \dots & \dots & \dots & \dots \\ \Delta x_{n+k-1} & x_{n+1} & \dots & x_{n+2k-2} \end{vmatrix}}$$

et donc

$$G_{k+1}(x_n, S_n) = \frac{\begin{vmatrix} S_n & S_{n+1} & \dots & S_{n+k+1} \\ x_n & x_{n+1} & \dots & x_{n+k+1} \\ \dots & \dots & \dots & \dots \\ x_{n+k} & x_{n+k+1} & \dots & x_{n+2k+1} \end{vmatrix}}{\begin{vmatrix} \Delta x_n & \Delta x_{n+1} & \dots & \Delta x_{n+k} \\ \Delta x_{n+1} & \dots & \dots & \Delta x_{n+k+1} \\ \dots & \dots & \dots & \dots \\ \Delta x_{n+k} & x_{n+k+1} & \dots & \Delta x_{n+2k} \end{vmatrix}} = \frac{\begin{vmatrix} S_n & x_n & \dots & x_{n+k} \\ S_{n+1} & x_{n+1} & \dots & x_{n+k+1} \\ \dots & \dots & \dots & \dots \\ S_{n+k+1} & x_{n+k+1} & \dots & x_{n+2k+1} \end{vmatrix}}{\begin{vmatrix} \Delta x_n & \Delta x_{n+1} & \dots & \Delta x_{n+k} \\ \Delta x_{n+1} & \Delta x_{n+2} & \dots & \Delta x_{n+k+1} \\ \dots & \dots & \dots & \dots \\ \Delta x_{n+k} & \Delta x_{n+k+1} & \dots & \Delta x_{n+2k} \end{vmatrix}}$$



Nous obtenons :

$$G_{k+1}(x_n, S_n) = q_{k+1}(x_n, S_n)$$

et si nous prenons la suite auxiliaire  $x_n = \Delta S_n$  nous avons

$$G_{k+1}(\Delta S_n, S_n) = \varepsilon_{2k+2}^{(n)} \quad \text{où } \varepsilon_{2k+2}^{(n)} \text{ sont les transformés de } S_n \text{ par}$$

l' $\varepsilon$ -algorithme.

Nous allons nous intéresser maintenant au calcul de  $G_{k+1}(x_n, S_n)$  et pour cela nous nous inspirons de l'algorithme de PYE et ATCHISON [42]. Dans cette étude nous avons besoin d'un certain nombre de notations :

- nous appellerons déterminant de Hankel d'ordre  $r$  le déterminant :

$$\begin{vmatrix} x_n & x_{n+1} & \dots & x_{n+r-1} \\ x_{n+1} & x_{n+2} & \dots & x_{n+r} \\ \dots & \dots & \dots & \dots \\ x_{n+r-1} & x_{n+r} & \dots & x_{n+2r-2} \end{vmatrix}$$

que nous noterons :

$$H_r^{(n)}[x_i] \quad \text{pour } r = 1, 2, \dots$$

avec

$$H_0^{(n)}[x_i] \equiv 1$$

- nous appellerons déterminant de Hankel bordé d'ordre  $r$  le déterminant :

$$\begin{vmatrix} u & v & z_n & z_{n+1} & \dots & z_{n+r-3} \\ w_n & y_n & x_n & x_{n+1} & \dots & x_{n+r-3} \\ w_{n+1} & y_{n+1} & x_{n+1} & x_{n+2} & \dots & x_{n+r-2} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ w_{n+r-2} & y_{n+r-2} & x_{n+r-2} & x_{n+r-1} & \dots & x_{n+2r-5} \end{vmatrix}$$

que nous noterons :

$$H_r^{(n)} \begin{bmatrix} u & v & z_i \\ w_i & y_i & [x_i] \end{bmatrix} \quad \text{pour } r \geq 3$$

Définition 8 : soient  $\{S_n\}$  et  $\{x_n\}$  deux suites de nombres réels alors :

$$G_0(x_n, S_n) = S_n \quad \forall n = 0, 1, 2$$

$$G_k(x_n, S_n) = \frac{H_{k+1}^{(n)} \begin{bmatrix} S_i \\ [x_i] \end{bmatrix}}{H_{k+1}^{(n)} \begin{bmatrix} 1 \\ [x_i] \end{bmatrix}} \quad \forall k = 1, 2, \dots$$

$$\forall n = 0, 1, 2, \dots$$

et

$$R_k(x_n, S_n) = \frac{H_k^{(n+1)} [x_i] H_k^{(n)} \begin{bmatrix} 1 \\ [x_i] \end{bmatrix}}{H_k^{(n)} [x_i] H_k^{(n+1)} \begin{bmatrix} 1 \\ [x_i] \end{bmatrix}} \quad \forall n = 0, 1, 2, \dots$$

$$\forall k = 0, 1, 2, \dots$$

Nous pouvons remarquer que pour  $k = 1$  nous obtenons

$$G_1(x_n, S_n) = S_n + \frac{S_{n+1} - S_n}{1 - \frac{x_{n+1}}{x_n}}$$

qui peut encore s'écrire, en introduisant  $R_k(x_n, S_n)$

$$G_1(x_n, S_n) = S_n + \frac{S_{n+1} - S_n}{1 - R_1(x_n, S_n)}$$

$$\text{où } G_1(x_n, S_n) = G_0(x_n, S_n) + \frac{G_0(x_{n+1}, S_{n+1}) - G_0(x_n, S_n)}{1 - R_1(x_n, S_n)}$$

De cette formule, l'idée est venue de généraliser pour  $k > 1$ . Avant cela nous avons besoin d'un résultat.

Lemme 1 : Soit  $A$  une matrice  $n \times n$   $c_1, c_2$  de vecteurs colonnes  $R_1, R_2$  deux vecteurs lignes alors

$$|A| \begin{vmatrix} A & c_1 & c_2 \\ R_1 & d_{11} & d_{12} \\ R_2 & d_{21} & d_{22} \end{vmatrix} = \begin{vmatrix} A & c_1 \\ R_1 & d_{11} \end{vmatrix} \begin{vmatrix} A & c_2 \\ R_2 & d_{22} \end{vmatrix} - \begin{vmatrix} A & c_2 \\ R_1 & d_{12} \end{vmatrix} \begin{vmatrix} A & c_1 \\ R_2 & d_{21} \end{vmatrix}$$

## IV.2 ALGORITHME DE CALCUL

Nous pouvons énoncer le résultat suivant :

Théorème 54 : si  $H_k^{(n+1)}[S_i]$ ,  $H_{k+1}^{(n)} \begin{bmatrix} 1 \\ [x_i] \end{bmatrix}$ ,  $H_{k+1}^{(n+1)} \begin{bmatrix} 1 \\ [x_i] \end{bmatrix}$  et

$H_{k+2}^{(n)} \begin{bmatrix} 1 \\ [x_i] \end{bmatrix}$  sont différents de zéro, alors :

$$G_{k+1}(x_n, S_n) = G_k(x_n, S_n) + \frac{G_k(x_{n+1}, S_{n+1}) - G_k(x_n, S_n)}{1 - R_{k+1}(x_n, S_n)}$$

Démonstration : pour  $k=0$  ceci est vraie pour  $k \geq 1$  nous appliquons le lemme 1

$$H_k^{(n+1)}[x_i] H_{k+2}^{(n)} \begin{bmatrix} S_i \\ [x_i] \end{bmatrix} = H_{k+1}^{(n+1)}[x_i] H_{k+1}^{(n)} \begin{bmatrix} S_i \\ [x_i] \end{bmatrix} - H_{k+1}^{(n)}[x_i] H_{k+1}^{(n+1)} \begin{bmatrix} S_i \\ [x_i] \end{bmatrix}$$

et

$$H_k^{(n+1)}[x_i] H_{k+2}^{(n)} \begin{bmatrix} S_i \\ [x_i] \end{bmatrix} = H_{k+1}^{(n+1)}[x_i] H_{k+1}^{(n)} \begin{bmatrix} 1 \\ [x_i] \end{bmatrix} - H_{k=1}^{(n)}[x_i] H_{k+1}^{(n+1)} \begin{bmatrix} 1 \\ [x_i] \end{bmatrix}$$

et en divisant membre à membre :

$$G_{k+1}(x_n, S_n) = \frac{R_{k+1}(x_n, S_n) G_k(x_n, S_n) - G_k(x_{n+1}, S_{n+1})}{R_{k+1}(x_n, S_n) - 1}$$

c.Q.F.D.

Il nous reste le problème de l'évaluation des  $R_k(x_n, S_n)$ . Pour cela il nous faut prendre de nouvelles notations. Posons :

$$r_k^{(n)} = \frac{H_k^{(n)}[x_i]}{H_k^{(n)} \begin{bmatrix} 1 \\ [x_i] \end{bmatrix}} \quad \forall k, n = 0, 1, \dots$$

$$s_k^{(n)} = \frac{H_{k+1}^{(n)} \begin{bmatrix} 1 \\ [x_i] \end{bmatrix}}{H_k^{(n)}[x_i]} \quad \forall k, n = 0, 1, \dots$$

Les  $R_k(x_n, S_n)$  peuvent être alors écrit  $R_k(x_n, S_n) = \frac{r_k^{(n+1)}}{r_k^{(n)}}$ .

Il nous faut maintenant donner un algorithme qui génère les quantités  $r_k^{(n)}$ . Pour cela nous utiliserons deux résultats.

Lemme 2 : Si  $k$  et  $n$  sont des entiers tels que  $k \geq 1$  et  $n \geq 0$  alors

$$H_{k-1}^{(n+1)}[x_i] H_{k+1}^{(n)} \begin{bmatrix} 1 \\ [x_i] \end{bmatrix} + H_k^{(n)}[x_i] H_k^{(n+1)} \begin{bmatrix} 1 \\ [x_i] \end{bmatrix} - H_k^{(n+1)}[x_i] H_k^{(n)} \begin{bmatrix} 1 \\ [x_i] \end{bmatrix} = 0$$

Lemme 3 : Si  $k$  et  $n$  sont des entiers non négatifs, alors :

$$H_k^{(n+1)}[x_i] H_{k+1}^{(n)} \begin{bmatrix} 1 \\ [x_i] \end{bmatrix} + H_{k+1}^{(n)}[x_i] H_k^{(n+1)} \begin{bmatrix} 1 \\ [x_i] \end{bmatrix} - H_k^{(n)}[x_i] H_{k+1}^{(n+1)} \begin{bmatrix} 1 \\ [x_i] \end{bmatrix} = 0$$

Démonstration : [46 ]

Les deux lemmes ci-dessus nous conduisent au résultat :

Théorème 55 : Nous avons la relation suivant :

$$s_k^{(n)} / s_{k-1}^{(n+1)} + 1 = r_k^{(n+1)} / r_k^{(n)}$$

$$s_k^{(n+1)} / s_k^{(n)} = r_{k+1}^{(n)} / r_k^{(n+1)} + 1$$

avec les conditions initiales :  $s_0^{(n)} = 1$ ,  $r_1^{(n)} = x_n$ .

Ce théorème nous donne un algorithme récursif permettant de calculer  $r_k^{(n)}$  donc les  $R_k^{(n)}$ .

### IV.3 VISUALISATION DE L'ALGORITHME

D'abord le calcul des  $r_k^{(n)}$  et  $s_k^{(n)}$  :

nous initialisons  $r_1^{(n)} = x_n$  au  $x_n$  est une suite auxiliaire  
 $s_0^{(n)} = 1$

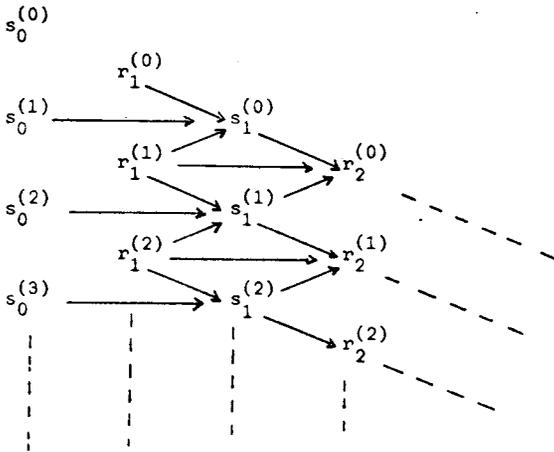
les  $r_k^{(n)}$  et  $s_k^{(n)}$  sont donnés par les relations :

$$s_k^{(n)} = \left( \frac{r_k^{(n+1)}}{r_k^{(n)}} - 1 \right) s_{k-1}^{(n+1)} \quad (E_1)$$

$$\forall k \geq 1 \quad r_{k+1}^{(n)} = \left( \frac{s_k^{(n+1)}}{s_k^{(n)}} - 1 \right) r_k^{(n+1)} \quad (E_2)$$

$$\forall n \geq 0$$

Cet algorithme est un algorithme en losange. Les relations  $(E_1)$  et  $(E_2)$  relient les quantités situées aux quatre sommets d'un losange : la quantité la plus à droite est calculée à partir des trois autres comme l'indiquent les flèches dans le schéma suivant :

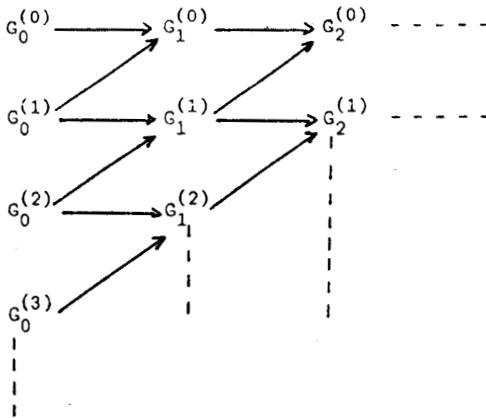


de cet algorithme, qui nous donne les  $r_k^{(n)}$  nous obtenons les  $R_k^{(n)} = \frac{r_k^{(n+1)}}{r_k^{(n)}}$

qui nous permettent de calculer les  $G_k^{(n)}$  à l'aide des relations :

$$\forall k \geq 0, \forall n \geq 0, G_{k+1}^{(n)} = G_k^{(n)} + \frac{G_k^{(n+1)} - G_k^{(n)}}{1 - R_{k+1}^{(n)}} \quad \text{avec } G_0^{(n)} = S_n$$

Les quantités  $G_k^{(n)}$  peuvent être rangées dans le tableau suivant :



#### IV.4 THÉORÈMES DE CONVERGENCE

Ce procédé, considéré sous cette nouvelle formulation peut être apparenté à d'autres procédés d'accélération. En effet pour  $k=1$

$$q_1^{(n)} = S_n + \frac{S_{n+1} - S_n}{1 - \frac{x_{n+1}}{x_n}} = \frac{x_n S_{n+1} - x_{n+1} S_n}{x_n - x_{n+1}}$$

qui est le procédé de Richardson [47] d'ordre 1.

De manière générale, nous pouvons écrire :

$$q_{k+1}^{(n)} = \frac{r_{k+1}^{(n+1)} q_k^{(n)} - r_{k+1}^{(n)} q_k^{(n+1)}}{r_k^{(n+1)} - r_k^{(n)}}$$

Ce procédé peut être considéré comme une certaine généralisation du procédé de Richardson où les éléments  $r_k^{(n)}$  sont calculés à chaque étape à l'aide des équations (E1) et (E2). Mais, d'une certaine manière, ce procédé généralise une classe de transformations non linéaires introduites par W.D. CLARK et H.L. GRAY [33] et étudiées par R.F. STREIT [51].

Nous pourrions comme pour les autres algorithmes, étudier le cas particulier de la première itération : la colonne  $\{q_1^{(n)}\}$ , mais comme nous l'avons vu ci-dessus, c'est la première étape du procédé de Richardson qui a déjà été très étudié : BREZINSKI [4], LAURENT [38].

Nous allons donc étudier les autres colonnes et en particulier le passage de la colonne  $k$  à la colonne  $k+1$  c'est-à-dire le passage de  $\{q_k^{(n)}\}$  à  $\{q_{k+1}^{(n)}\}$ . Soit  $\{S_n\}$  une suite de nombres réels de limite  $S$  ; nous lui appliquons le procédé  $q$  associé à la suite auxiliaire  $x_n$ .

Avant d'étudier ce procédé  $q$  à l'aide de la nouvelle formulation, nous allons citer quelques propriétés et théorèmes [4] tirés de la définition : rapport de deux déterminants :

$$q_{k+1}(x_n, S_n) = \frac{\begin{vmatrix} S_n & x_n & \dots & x_{n+k} \\ \dots & \dots & \dots & \dots \\ S_{n+k+1} & x_{n+k+1} & \dots & x_{n+2k+1} \end{vmatrix}}{\begin{vmatrix} \Delta x_n & \dots & \dots & \Delta x_{n+k} \\ \dots & \dots & \dots & \dots \\ \Delta x_{n+k} & \dots & \dots & \Delta x_{n+2k} \end{vmatrix}}$$

BOIS  
LILLE

Théorème 56 : une condition nécessaire et suffisante pour que

$q_{k+1}(x_n, S_n) = S, \forall n > N$  est que la suite  $\{S_n\}$  vérifie

$$S_n - S = \sum_{i=0}^k a_i x_{n+i}$$

Théorème 57 :

$$- \text{ Si } S_n - S = \sum_{i=0}^k a_i^{(n)} x_{n+i} \quad \text{pour } n = 0, 1, \dots$$

$$- \text{ Si } \lim_{n \rightarrow \infty} a_i^{(n)} = a_i \quad \text{pour } i = 0, \dots, k$$

$$\text{alors } \lim_{n \rightarrow \infty} q_{k+1}^{(n)}(x_n, S_n) = S.$$

Propriété 8 :

$$q_k(ax_n, cS_n + d) = cq_k(x_n, S_n) + d$$

$$q_{k+1}(\Delta S_n, S_n) = \epsilon_{2k+2}^{(n)}$$

Lorsqu'il n'y a pas ambiguïté, nous noterons les éléments  $q_k(x_n, S_n) = q_k^{(n)}$ .

Nous étudions maintenant les conditions de régularité.

Théorème 58 : si  $\{q_k^{(n)}\}$  converge, une condition nécessaire et

suffisante pour que  $\lim_{n \rightarrow \infty} q_{k+1}^{(n)} = \lim_{n \rightarrow \infty} q_k^{(n)}$  est que :

$$\lim_{n \rightarrow \infty} \frac{\Delta q_k^{(n)}}{1 - R_{k+1}^{(n)}} = 0$$

Démonstration :

$$q_{k+1}^{(n)} = q_k^{(n)} + \frac{\Delta q_k^{(n)}}{1 - R_{k+1}^{(n)}} \quad \forall n$$

$$|q_{k+1}^{(n)} - q_k^{(n)}| = \left| \frac{\Delta q_k^{(n)}}{1 - R_{k+1}^{(n)}} \right|$$

d'où  $\lim_{n \rightarrow \infty} |q_{k+1}^{(n)} - q_k^{(n)}| = 0$  et  $\lim_{n \rightarrow \infty} q_{k+1}^{(n)} = \lim_{n \rightarrow \infty} q_k^{(n)}$

la réciproque est immédiate.

C.Q.F.D.

Théorème 59 : Si  $\{q_k^{(n)}\}$  converge et si il existe une constante  $c > 0$  telle que  $|1 - R_{k+1}^{(n)}| > c, \forall n > N$  alors  $\lim_{n \rightarrow \infty} q_{k+1}^{(n)} = \lim_{n \rightarrow \infty} q_k^{(n)}$ .

Démonstration :

$$|q_{k+1}^{(n)} - q_k^{(n)}| = \left| \frac{\Delta q_k^{(n)}}{1 - R_{k+1}^{(n)}} \right|$$

$$|q_{k+1}^{(n)} - q_k^{(n)}| < c^{-1} |\Delta q_k^{(n)}|$$

puisque  $q_k^{(n)}$  converge  $\lim_{n \rightarrow \infty} |\Delta q_k^{(n)}| = 0$

donc  $\lim_{n \rightarrow \infty} |q_{k+1}^{(n)} - q_k^{(n)}| = 0$ .

C.Q.F.D.

Corollaire : si  $\{q_k^{(n)}\}$  converge et si  $\lim_{n \rightarrow \infty} R_{k+1}^{(n)} = R_{k+1} \neq 1$  alors

$$\lim_{n \rightarrow \infty} q_{k+1}^{(n)} = \lim_{n \rightarrow \infty} q_k^{(n)} .$$

Démonstration : en effet si  $\lim_{n \rightarrow \infty} R_{k+1}^{(n)} \neq 1$  alors il existe  $c > 0$

telle que  $\forall n > N \quad |R_{k+1}^{(n)} - 1| > c$  en appliquant le théorème précédent nous obtenons le résultat.

Théorème 60 : si  $\{q_k^{(n)}\}$  est monotone convergente et si

$$\lim_{n \rightarrow \infty} n(1 - R_{k+1}^{(n)}) = c \neq 0 \text{ alors : } \lim_{n \rightarrow \infty} q_{k+1}^{(n)} = \lim_{n \rightarrow \infty} q_k^{(n)} .$$

Démonstration :

$$q_{k+1}^{(n)} - q_k^{(n)} = \frac{\Delta q_k^{(n)}}{1 - R_{k+1}^{(n)}}$$

$q_k^{(n)}$  est convergente monotone alors  $\lim_{n \rightarrow \infty} n \Delta q_k^{(n)} = 0$

$$\frac{\Delta q_k^{(n)}}{1 - R_{k+1}^{(n)}} = \frac{n \Delta q_k^{(n)}}{n(1 - R_{k+1}^{(n)})} \xrightarrow{n \rightarrow \infty} 0$$

donc  $\lim_{n \rightarrow \infty} q_{k+1}^{(n)} = \lim_{n \rightarrow \infty} q_k^{(n)}$ .

Corollaire : si  $q_k^{(n)}$  est convergente et monotone et si  $r_{k+1}^{(n)}$  est telle que  $\sum_{i=0}^n r_{k+1}^{(i)}$  converge : alors  $\lim_{n \rightarrow \infty} q_{k+1}^{(n)} = \lim_{n \rightarrow \infty} q_k^{(n)}$ .

Démonstration : d'après Raabe,  $\sum_{i=0}^n r_{k+1}^{(i)}$  converge alors  $n(1 - \frac{r_{k+1}^{(n+1)}}{r_{k+1}^{(n)}})$

ne converge pas vers zéro, d'où le résultat.

Nous allons nous intéresser maintenant au problème de l'accélération de la convergence. Pour cela nous devons nous placer sous certaines hypothèses d'existence de limites : nous nous plaçons à l'étape  $k$  et nous supposons que :

- $q_k^{(n)}$  converge vers  $S$
- $\lim_{n \rightarrow \infty} \frac{q_k^{(n+1)} - S}{q_k^{(n)} - S}$  existe
- $\lim_{n \rightarrow \infty} \frac{r_{k+1}^{(n+1)}}{r_{k+1}^{(n)}}$  existe.

Dans ces hypothèses, nous pouvons énoncer un premier résultat.

Théorème 61 : si  $\lim_{n \rightarrow \infty} q_{k+1}^{(n)} = \lim_{n \rightarrow \infty} q_k^{(n)}$ , alors :

$$\left\{ \lim_{n \rightarrow \infty} \frac{q_k^{(n+1)} - s}{q_k^{(n)} - s} = \lim_{n \rightarrow \infty} \frac{r_{k+1}^{(n+1)}}{r_{k+1}^{(n)}} \neq 1, 0 \right\} \iff \left\{ q_{k+1}^{(n)} - s = o(q_k^{(n+1)} - s) \right\}$$

Démonstration :

$$\frac{q_k^{(n)} - s + \frac{q_k^{(n+1)} - q_k^{(n)}}{1 - \frac{r_{k+1}^{(n+1)}}{r_{k+1}^{(n)}}}}{q_k^{(n+1)} - s} = \frac{q_k^{(n+1)} - q_k^{(n)}}{q_k^{(n+1)} - s}$$

$$\frac{q_{k+1}^{(n)} - s}{q_k^{(n+1)} - s} = \frac{q_k^{(n)} - s}{q_k^{(n+1)} - s} + \frac{1 - \frac{q_k^{(n)} - s}{q_k^{(n+1)} - s}}{1 - \frac{r_{k+1}^{(n+1)}}{r_{k+1}^{(n)}}}$$

$$\frac{q_{k+1}^{(n)} - s}{q_k^{(n+1)} - s} = \frac{q_k^{(n)} - s}{q_k^{(n+1)} - s} \left[ 1 - \frac{1}{1 - \frac{r_{k+1}^{(n+1)}}{r_{k+1}^{(n)}}} \right] + \frac{1}{1 - \frac{r_{k+1}^{(n+1)}}{r_{k+1}^{(n)}}}$$

$$\frac{q_{k+1}^{(n)} - s}{q_k^{(n+1)} - s} = \frac{q_k^{(n)} - s}{q_k^{(n+1)} - s} \left[ - \frac{\frac{r_{k+1}^{(n+1)}}{r_{k+1}^{(n)}}}{1 - \frac{r_{k+1}^{(n+1)}}{r_{k+1}^{(n)}}} \right] + \frac{1}{1 - \frac{r_{k+1}^{(n+1)}}{r_{k+1}^{(n)}}}$$



notons 
$$Q_k = \lim_{n \rightarrow \infty} \frac{q_k^{(n+1)} - S}{q_k^{(n)} - S}$$

et 
$$R_{k+1} = \lim_{n \rightarrow \infty} \frac{r_{k+1}^{(n+1)}}{r_{k+1}^{(n)}}$$

alors

$$\lim_{n \rightarrow \infty} \frac{q_{k+1}^{(n)} - S}{q_k^{(n+1)} - S} = \frac{1}{Q_k} \left[ - \frac{R_{k+1}}{1 - R_{k+1}} \right] + \frac{1}{1 - R_{k+1}}$$

d'où le résultat

$$\lim_{n \rightarrow \infty} \frac{q_{k+1}^{(n)} - S}{q_k^{(n+1)} - S} = 0 \iff R_{k+1} = Q_k$$

Là également, il n'est pas nécessaire que la suite converge pour parler d'accélération de convergence ; cependant ici nous devons prendre une hypothèse supplémentaire :

$$\lim_{n \rightarrow \infty} \frac{\Delta q_k^{(n+1)}}{\Delta q_k^{(n)}} \text{ existe}$$

Théorème 62 :

$$\left\{ \lim_{n \rightarrow \infty} \frac{\Delta q_k^{(n+1)}}{\Delta q_k^{(n)}} = \lim_{n \rightarrow \infty} \frac{r_{k+1}^{(n+1)}}{r_{k+1}^{(n)}} \neq 0, 1 \right\} \iff \left\{ \Delta q_{k+1}^{(n)} = o(\Delta q_k^{(n+1)}) \right\}$$

Démonstration :

$$\frac{\Delta q_k^{(n)}}{\Delta q_k^{(n+1)}} = \frac{\Delta q_k^{(n)} + \frac{\Delta q_k^{(n+1)}}{1 - \frac{r_{k+1}^{(n+2)}}{r_{k+1}^{(n+1)}}} - \frac{\Delta q_k^{(n)}}{1 - \frac{r_{k+1}^{(n+1)}}{r_{k+1}^{(n)}}}}{\Delta q_k^{(n+1)}}$$

$$\frac{\Delta q_{k+1}^{(n)}}{\Delta q_k^{(n+1)}} = \frac{\Delta q_k^{(n)}}{\Delta q_k^{(n+1)}} + \frac{1}{1 - \frac{r_{k+1}^{(n+2)}}{r_{k+1}^{(n+1)}}} - \frac{\frac{\Delta q_k^{(n)}}{\Delta q_k^{(n+1)}}}{1 - \frac{r_{k+1}^{(n+1)}}{r_{k+1}^{(n)}}}$$

$$\frac{\Delta q_{k+1}^{(n)}}{\Delta q_k^{(n+1)}} = \frac{\Delta q_k^{(n)}}{\Delta q_k^{(n+1)}} \left[ 1 - \frac{1}{1 - \frac{r_{k+1}^{(n+1)}}{r_{k+1}^{(n)}}} \right] + \frac{1}{1 - \frac{r_{k+1}^{(n+2)}}{r_{k+1}^{(n+1)}}}$$

$$\frac{\Delta q_{k+1}^{(n)}}{\Delta q_k^{(n+1)}} = \frac{\Delta q_k^{(n)}}{\Delta q_k^{(n+1)}} \left[ - \frac{\frac{r_{k+1}^{(r+1)}}{r_{k+1}^{(n)}}}{1 - \frac{r_{k+1}^{(n+1)}}{r_{k+1}^{(n)}}} \right] + \frac{1}{1 - \frac{r_{k+1}^{(n+2)}}{r_{k+1}^{(n+1)}}}$$



$$\text{posons : } \lim_{n \rightarrow \infty} \frac{\Delta q_k^{(n+1)}}{\Delta q_k^{(n)}} = Q'_k$$

$$\lim_{n \rightarrow \infty} \frac{r_{k+1}^{(n+1)}}{r_{k+1}^{(n)}} = R_{k+1}$$

$$\lim_{n \rightarrow \infty} \frac{\Delta q_{k+1}^{(n)}}{\Delta q_k^{(n+1)}} = \frac{1}{Q'_k} \left[ - \frac{R_{k+1}}{1 - R_{k+1}} \right] + \frac{1}{1 - R_{k+1}}$$

d'où le résultat

$$\lim_{n \rightarrow \infty} \frac{\Delta q_{k+1}^{(n)}}{\Delta q_k^{(n+1)}} = 0 \iff R_{k+1} = Q'_k.$$

Ces conditions d'accélération sont difficilement vérifiables dès que  $k \geq 2$  cependant nous pouvons étudier le comportement des rapports

$$\frac{r_k^{(n+1)}}{r_k^{(n)}} ; \text{ nous avons les relations :}$$

$$\frac{r_{k+1}^{(n+1)}}{r_{k+1}^{(n)}} = \begin{bmatrix} \frac{s_k^{(n+2)}}{s_k^{(n+1)}} - 1 \\ \frac{s_k^{(n+1)}}{s_k^{(n)}} - 1 \end{bmatrix} \frac{r_k^{(n+2)}}{r_k^{(n+1)}} ; \frac{s_{k+1}^{(n+1)}}{s_{k+1}^{(n)}} = \begin{bmatrix} \frac{r_{k+1}^{(n+2)}}{r_{k+1}^{(n+1)}} - 1 \\ \frac{r_{k+1}^{(n+1)}}{r_{k+1}^{(n)}} - 1 \end{bmatrix} \frac{s_k^{(n+2)}}{s_k^{(n+1)}}$$

pour  $k=0$  nous avons

$$s_0^{(n)} = 1 \quad \forall n$$

$$r_1^{(n)} = x_n$$

$$s_1^{(n)} = \frac{\Delta x_n}{x_n}$$

Il y a deux cas :

$$1) . \lim_{n \rightarrow \infty} \frac{x_{n+1}}{x_n} = A \neq 1 \text{ alors :}$$

$$- \lim_{n \rightarrow \infty} \frac{s_1^{(n+1)}}{s_1^{(n)}} = 1 \implies \frac{r_2^{(n)}}{r_1^{(n+1)}} \xrightarrow{n \rightarrow \infty} 0$$

donc si  $x_n$  converge vers 0

$r_2^{(n)}$  converge plus vite que  $x_{n+1}$

$$- \lim_{n \rightarrow \infty} \frac{r_2^{(n+1)}}{r_2^{(n)}} = \lim_{n \rightarrow \infty} \frac{r_1^{(n+1)}}{r_1^{(n)}} \cdot \lim_{n \rightarrow \infty} \frac{\Delta s_1^{(n+1)}}{\Delta s_1^{(n)}}$$

si ces limites existent alors  $\lim_{n \rightarrow \infty} \frac{r_2^{(n+1)}}{r_2^{(n)}} \neq 1$

$$2) . \lim_{n \rightarrow \infty} \frac{x_{n+1}}{x_n} = 1 \text{ alors}$$

$$\lim_{n \rightarrow \infty} \frac{s_1^{(n+1)}}{s_1^{(n)}} = 1$$

$$\lim_{n \rightarrow \infty} \frac{s_1^{(n+1)}}{s_1^{(n)}} = 1$$

$$\implies \lim_{n \rightarrow \infty} \frac{r_2^{(n)}}{r_1^{(n+1)}} = 0$$

$$\implies \lim_{n \rightarrow \infty} \frac{r_2^{(n+1)}}{r_2^{(n)}} = \lim_{n \rightarrow \infty} \frac{\Delta s_1^{(n+1)}}{\Delta s_1^{(n)}}$$

$$\lim_{n \rightarrow \infty} \frac{\Delta r_1^{(n+1)}}{\Delta r_1^{(n)}} \quad \text{là il y a 2 cas}$$

$$\lim_{n \rightarrow \infty} \frac{s_1^{(n+1)}}{s_1^{(n)}} \neq 1$$

$$\implies \lim_{n \rightarrow \infty} \frac{r_2^{(n+1)}}{r_2^{(n)}} = 1$$

Nous pouvons généraliser l'étude à l'étape k, supposons que

$$\lim_{n \rightarrow \infty} \frac{s_k^{(n+1)}}{s_k^{(n)}} = S_k$$

$$\lim_{n \rightarrow \infty} \frac{r_{k+1}^{(n+1)}}{r_{k+1}^{(n)}} = R_{k+1}$$



Il y a deux cas à envisager :

1) .  $R_{k+1} \neq 1$  alors

$$\lim_{n \rightarrow \infty} \frac{s_{k+1}^{(n+1)}}{s_{k+1}^{(n)}} = S_k$$

$S_k = 1$

$\Rightarrow \lim_{n \rightarrow \infty} \frac{r_{k+2}^{(n)}}{r_{k+1}^{(n+1)}} = 0$

$\Rightarrow \lim_{n \rightarrow \infty} \frac{r_{k+2}^{(n+1)}}{r_{k+2}^{(n)}} = R_{k+1} \cdot \lim_{n \rightarrow \infty} \frac{\Delta s_{k+1}^{(n+1)}}{\Delta s_{k+1}^{(n)}}$

$S_k \neq 1$

$\Rightarrow \lim_{n \rightarrow \infty} \frac{r_{k+2}^{(n+1)}}{r_{k+2}^{(n)}} = R_{k+1}$

si les limites existent

2) .  $R_{k+1} = 1$  alors

$$\lim_{n \rightarrow \infty} \frac{s_{k+1}^{(n+1)}}{s_{k+1}^{(n)}} = S_k \lim_{n \rightarrow \infty} \frac{\Delta r_{k+1}^{(n+1)}}{\Delta r_{k+1}^{(n)}}$$

$\lim_{n \rightarrow \infty} \frac{s_{k+1}^{(n+1)}}{s_{k+1}^{(n)}} = 1$

$\Rightarrow \lim_{n \rightarrow \infty} \frac{r_{k+2}^{(n)}}{r_{k+1}^{(n+1)}} = 0$

$\Rightarrow \lim_{n \rightarrow \infty} \frac{r_{k+2}^{(n+1)}}{r_{k+2}^{(n)}} = R_{k+1} \lim_{n \rightarrow \infty} \frac{\Delta s_{k+1}^{(n+1)}}{\Delta s_{k+1}^{(n)}}$

$\lim_{n \rightarrow \infty} \frac{s_{k+1}^{(n+1)}}{s_{k+1}^{(n)}} \neq 1$

$\Rightarrow \lim_{n \rightarrow \infty} \frac{r_{k+2}^{(n+1)}}{r_{k+2}^{(n)}} = R_{k+1} = 1$

Il est donc possible de connaître le comportement des suites auxiliaires ; mais la difficulté majeure vient de la détermination de

$\lim_{n \rightarrow \infty} \frac{\Delta q_k^{(n+1)}}{\Delta q_k^{(n)}}$  qui se met sous la forme % et l'indétermination est difficile

à lever.

Pendant voici quelques résultats encourageants : nous comparons les résultats obtenus avec l' $\epsilon$ -algorithme et le procédé q ; dans les tableaux n désigne le nombre de termes de la suite initiale nécessaire au calcul des éléments.

Dans ces exemples nous comparons les éléments de la diagonale principale c'est-à-dire  $\epsilon_{2n}^{(0)}$  et  $q_n^{(0)}$ .

Exemple 1 :

$$S_n = \sum_{k=1}^n (-1)^{k+1} / (2k-1) \quad S = 0,7853981633974$$

$$r_n = (-1)^n \log(1+1/n)$$

n	$S_n$	$\epsilon$ algorithme	procédé q
1	1.0	1.0	1.0
3	<u>0.86</u>	<u>0.79</u>	<u>0.785</u>
5	<u>0.83</u>	<u>0.7855</u>	<u>0.785393</u>
7	<u>0.82</u>	<u>0.7854</u>	<u>0.7853980</u>
9	<u>0.81</u>	<u>0.7853983</u>	<u>0.785398162</u>
11	<u>0.808</u>	<u>0.785398168</u>	<u>0.78539816337</u>
13	<u>0.804</u>	<u>0.7853981635</u>	<u>0.785398163393</u>

Exemple 2 :

$$S_n = (0,98)^n \qquad S = 0$$

$$x = 0,2 (0,98)^{n-1} \qquad 0$$

n	$S_n$	$\epsilon$ algorithme	procédé q
1	0,98	0,98	0,98
3	0,94	$0,2120 \times 10^{-13}$	$0,5551 \times 10^{-15}$
5	0,90	$0,1925 \times 10^{-13}$	$0,5624 \times 10^{-14}$

Sur ces deux premiers exemples nous pouvons remarquer la puissance du procédé q. Nous avons pris la suite auxiliaire liée à la suite initiale telle que :

$$\frac{\Delta S_{n+1}}{\Delta S_n} \text{ se comporte comme } \frac{x_{n+1}}{x_n}$$

Exemple 3 :

$$S_n = \sum_{k=1}^n (-1)^k / (1+k) \qquad S = \text{Log } 2 = 0,69314718055994$$

$$x_n = (-1)^n \text{Log } (1+1/n)$$

n	$\epsilon$ algorithme	procédé q
1	1.0	1.0
3	<u>0.70</u>	<u>0.692</u>
5	<u>0.6933</u>	<u>0.693143</u>
7	<u>0.69315</u>	<u>0.69314715</u>
9	<u>0.6931473</u>	<u>0.6931471802</u>
11	<u>0.693147184</u>	<u>0.69314718055</u>
13	<u>0.6931471806</u>	<u>0.6931471805592</u>
15	<u>0.69314718056</u>	<u>0.6931471805598</u>

Nous allons maintenant voir un certain nombre d'exemples où la suite  $S_n$  est une suite à convergence logarithmique c'est-à-dire au

$$\lim_{n \rightarrow \infty} \frac{\Delta S_{n+1}}{\Delta S_n} = 1.$$

Exemple 4 :

$$S_n = n/n+1$$

$$S = 1$$

$$x_n = 1/n$$

n	$S_n$	$\epsilon$ algorithme	procédé q
1	0.5	0.5	0.5
3	0.75	<u>0.83</u>	<u>1.00000000000000</u>
5	0.83	<u>0.916</u>	<u>1.00000000000000</u>
7	0.875	<u>0.95</u>	<u>1.000000000000007</u>
9	<u>0.9</u>	<u>0.96</u>	<u>1.000000000000037</u>
11	<u>0.916</u>	<u>0.97</u>	<u>1.000000000000176</u>
13	<u>0.928</u>	<u>0.98</u>	<u>0.999999999991639</u>

Exemple 5 :

$$S_{n-2} = \sum_{k=2}^n 1/(k^2-1) \quad n > 1 \quad S = 0.75$$

$$x_n = 1/n$$

n	$S_n$	$\epsilon$ algorithme	procédé q
1	0.33	0.33	0.33
3	0.525	0.60	0.733
5	0.595	0.67	0.750000000000001
7	0.631	0.70	0.750000000000007
9	0.654	0.71	0.750000000000034
11	0.669	0.72	0.750000000001545
13	0.680	0.73	0.75000000028483

Exemple 6 :

$$S_n = \sum_{i=1}^n 1/i^2$$

$$S = 1.644934066845$$

$$x_n = 1/n$$

n	$S_n$	$\epsilon$ algorithme	procédé q
1	<u>1.0</u>	<u>1.0</u>	<u>1</u>
3	<u>1.361</u>	<u>1.45</u>	<u>1.66</u>
5	<u>1.463</u>	<u>1.55</u>	<u>1.647</u>
7	<u>1.511</u>	<u>1.59</u>	<u>1.6454</u>
9	<u>1.539</u>	<u>1.60</u>	<u>1.6451</u>
11	<u>1.558</u>	<u>1.61</u>	<u>1.6450</u>
13	<u>1.570</u>	<u>1.62</u>	<u>1.64499</u>
15	<u>1.580</u>	<u>1.63</u>	<u>1.644951</u>

Cependant il ne faut pas croire que le procédé q est toujours meilleur que l' $\epsilon$ -algorithme. En effet le choix de la suite  $\{x_n\}$  est difficile et dans certains cas, bien que cette suite vérifie les conditions de convergence et d'accélération de la convergence, les résultats obtenus sont moins bons que pour l' $\epsilon$ -algorithme. De plus, pour certains des exemples précédents le  $\rho$  algorithme et le  $\theta$  algorithme fournissent des résultats meilleurs (pour cela on pourra consulter [68]).

## IV.5 APPLICATIONS

IV.5.1 INTEGRALES IMPROPRES DU PREMIER ORDRE

Cet algorithme a été calqué sur le  $G_m$  algorithme introduit par PYE et ATCHISON. Ce  $G_m$  algorithme est en fait le procédé q où nous avons fait les initialisations suivantes :

Soit  $f$  une fonction intégrable au sens de Riemann sur  $[a, \infty]$ ; pour  $k > 0$  nous notons :

$$F_n = \int_a^{t+nk} f(x) dx = F(t + nk)$$

$$f_n = f(t + nk)$$

la transformation  $G_n$  est définie comme suit :

$$G_0[F(t), k] \equiv F(t)$$

et  $\forall n > 0$

$$\left| \begin{array}{cccc} F(t) & F(t+k) & \dots & F(t+nk) \\ f(t) & f(t+k) & \dots & f(t+nk) \\ \dots & \dots & \dots & \dots \\ f(t+nk-k) & f(t+nk) & \dots & f(t+2nk-k) \end{array} \right|$$

$$G_n[F(t), k] = \frac{\left| \begin{array}{cccc} 1 & 1 & \dots & 1 \\ f(t) & f(t+k) & \dots & f(t+nk) \\ \dots & \dots & \dots & \dots \\ f(t+nk-k) & f(t+nk) & \dots & f(t+2nk-k) \end{array} \right|}{\left| \begin{array}{cccc} 1 & 1 & \dots & 1 \\ f(t) & f(t+k) & \dots & f(t+nk) \\ \dots & \dots & \dots & \dots \\ f(t+nk-k) & f(t+nk) & \dots & f(t+2nk-k) \end{array} \right|}$$

alors en prenant  $S_n = F_n$  et  $x_n = f_n$  dans le procédé  $q$ , nous avons  $G_n[F(t), k] = q_n(x_0, S_0)$  où les  $G_n[F(t), k]$  sont calculés récursivement à l'aide du  $G_n$  algorithme.

Le but principal de la transformation  $G_n$  est l'évaluation d'intégrales impropres du premier ordre. Nous n'allons étudier ici en détail cette transformation, ceci a été fait dans [29,30,31,32,33,34,35,36]. Cependant nous pouvons dire que les résultats de convergence énoncés au paragraphe précédent sont valables lorsque  $S_n = F_n$  et  $x_n = f_n$  et peuvent s'appliquer à la transformation  $G_n$  en particulier :

Théorème 63 :

$$- \text{ Si } -1 \leq \lim_{n \rightarrow \infty} \frac{f_{n+1}}{f_n} < 1$$

$$- \text{ Si } F_n \text{ converge quand } n \rightarrow \infty$$

$$- \text{ Si } \lim_{n \rightarrow \infty} \frac{r_i^{(n+1)}}{r_i^{(n)}} \text{ existe } \quad \forall i \leq k$$

$$- \text{ Si } |\Delta s_{i-1}^{(n+1)}| \leq |\Delta s_{i-1}^{(n)}| \quad \forall n > N \text{ et } \forall i \leq k$$

$$\text{alors } \lim_{n \rightarrow \infty} G_k[F(t), p] = \lim_{n \rightarrow \infty} F_n \quad \forall p > 0.$$

La démonstration est évidente en regardant au paragraphe précédent le comportement de  $R_i^{(n)}$ . Nous avons

$$\forall i \leq k \quad - 1 \leq \lim_{n \rightarrow \infty} R_i^{(n)} < 1.$$

Théorème 64 : Si  $\lim_{n \rightarrow \infty} \frac{f_{n+1}}{f_n} \neq 1, 0$  alors  $G_1[F(t), p]$  converge plus vite que  $F_{n+1}$ .

Démonstration : d'après les résultats du paragraphe précédent, il suffit de montrer que :

$$\lim_{n \rightarrow \infty} \frac{F_{n+1} - S}{F_n - S} = \lim_{n \rightarrow \infty} \frac{f_{n+1}}{f_n}$$

$$\lim_{n \rightarrow \infty} \frac{F_{n+1} - S}{F_n - S} = \lim_{n \rightarrow \infty} \frac{S - F_{n+1}}{S - F_n} = \lim_{n \rightarrow \infty} \frac{\int_{t+(n+1)p}^{\infty} f(x) dx}{\int_{t+np}^{\infty} f(x) dx}$$

en appliquant la règle de l'hospital :

$$\lim_{n \rightarrow \infty} \frac{F_{n+1} - S}{F_n - S} = \lim_{n \rightarrow \infty} \frac{f_{n+1}}{f_n}$$

C.Q.F.D.

Cette transformation est très utile en particulier pour l'évaluation de la transformée de Laplace d'une fonction, comme le montrent les résultats suivants :

Nous définissons l'opérateur L appliqué à la fonction g tel que :

$$L^n[g, a_i] = g^{(n)} + a_{n-1} g^{(n-1)} + \dots + a_1 g' + a_0$$

où les  $a_i$  sont des constantes.

Théorème 65 : Si  $L^n[g, a_i] = 0, \forall t \in [a, \infty]$  et si  $\lim_{t \rightarrow \infty} F(t) = S$  alors  $G_m[F(t), p] \equiv S, \forall p > 0, \forall m \geq n$ .

Corollaire : si  $L^n[f, a_1] = 0$  et si  $L[f, s]$  est la transformée de Laplace de paramètre  $s$  alors quel que soit  $s$  tel que  $L[f, s]$  existe :

$$G_n \left[ \int_0^t e^{-sx} f(x) dx, p \right] = L[f, s] \quad \forall t \geq 0$$

Ce corollaire est intéressant, en effet si  $f$  vérifie  $L^n[f, a_1] = 0$  (c'est-à-dire si  $f$  est solution d'une équation différentielle d'ordre  $n$ ) et si  $f$  est connu en certains points on peut donner sa transformée de Laplace.

Nous allons maintenant donner quelques exemples numériques. Dans ces exemples, nous évaluons  $F(t)$ ,  $F(t+p)$ ,... à l'aide d'une formule de Simpson dont la précision est en  $10^{-6}$  (pour plus de précision nous pourrions prendre Romberg...)

Exemple 1 :

$$F(t) = \int_0^t x e^{-x} dx \quad S = 1$$

$t$	$G_1[F(t), 1]$	$G_2[F(t), 1]$
6	<u>1.0015</u>	<u>0.999999997</u>
8	<u>1.00013</u>	"
10	<u>1.000013</u>	"
12	<u>1.0000014</u>	"
14	<u>1.00000016</u>	"
16	<u>1.000000017</u>	"

Exemple 2 : pour cet exemple  $f$  indique simplement les erreurs obtenues

$$F(t) = \int_0^t \frac{\sin x}{x} dx \quad S = 1$$

t	erreurs pour la transformation G
1	0.2
3	- 0.5005 $\times 10^{-4}$
5	- 0.1397 $\times 10^{-4}$
7	- 0,2662 $\times 10^{-6}$
9	- 0.4933 $\times 10^{-8}$

Cette transformation a été comparée [44] à la première forme confluyente de l' $\epsilon$ -algorithme [3,66]. Nous avons en général trouvé de meilleurs résultats numériques pour la transformation  $G_n$ , de plus elle ne nécessite pas le calcul des dérivées de la fonction  $f$  comme cela est le cas pour la forme confluyente de l' $\epsilon$ -algorithme. Cependant la forme confluyente de l' $\epsilon$ -algorithme ne nécessite que l'évolution d'une seule intégrale finie alors que pour la transformation  $G_n$  il faut en évaluer  $n$ .

#### IV.5.2 RELATION AVEC L' $\epsilon$ -ALGORITHME

Nous avons, au début du chapitre, remarqué que  $q_k(\Delta S_n S_n) = \epsilon_{2k}^{(n)}$ . Nous pouvons donc nous demander si l'un des deux algorithmes n'est pas plus performant que l'autre.

Une étude numérique faite dans [ ] nous laisse penser que ces deux algorithmes fournissent une précision comparable : lorsque l'un est instable, l'autre l'est également et réciproquement. Cependant du point de vue performance l' $\epsilon$ -algorithme est préférable :

- pour calculer  $\epsilon_{2k}^{(0)}$  il faut :
  - .  $k(2k+1)$  additions
  - .  $k(2k+1)$  soustractions
  - .  $k(2k+1)$  divisions

- pour calculer  $q_k(\Delta S_0, S_0)$  il faut :
  - .  $\frac{k(k+1)}{2}$  additions
  - .  $3k^2$  soustractions
  - .  $3k^2$  divisions
  - .  $(2k-1)k$  multiplications

De plus les résultats de convergence peuvent être appliqués au cas où  $x_n = \Delta S_n$ . Nous retrouvons des résultats connus :

Théorème 66 : si  $\lim_{n \rightarrow \infty} \frac{S_{n+1} - S}{S_n - S} \neq 1.0$  alors  $\{\epsilon_2^{(n)}\}$  converge plus vite que  $\{S_{n+1}\}$ .

Théorème 67 : si  $\{\epsilon_2^{(n)}\}$  converge alors  $\lim_{n \rightarrow \infty} \epsilon_2^{(n)} = \lim_{n \rightarrow \infty} S_n$ .

Nous pouvons également montrer un important résultat de convergence :

Théorème 68 : soit  $\{S_n\}$  une suite convergente de limite  $S$ . Si pour  $k$  fixé  $\{\epsilon_{2k}^{(n)}\}$  est convergente monotone de limite  $S$  alors si  $\{\epsilon_{2k+2}^{(n)}\}$  converge, elle converge vers la même limite que  $\{\epsilon_{2k}^{(n)}\}$ .

Démonstration : par identification avec le procédé  $q$  où  $x_n = \Delta S_n$ , nous pouvons écrire

$$\epsilon_{2k+2}^{(n)} = q_{k+1}^{(n)} = q_k^{(n)} + \frac{\Delta q_k^{(n)}}{1 - \frac{r_{k+1}^{(n+1)}}{r_{k+1}^{(n)}}}$$

Supposons que  $q_{k+1}^{(n)}$  converge vers  $T \neq S$ , ceci entraîne que

$$\lim_{n \rightarrow \infty} \frac{r_{k+1}^{(n+1)}}{r_{k+1}^{(n)}} = 1 \text{ et donc qu'il existe un rang } N \text{ à partir duquel } r_{k+1}^{(n)} > 0.$$

Etudions maintenant la suite  $r_{k+1}^{(n)}$ , nous avons  $\forall k \geq 0 \frac{r_{k+1}^{(n)}}{r_k^{(n+1)}} = \frac{s_k^{(n+1)}}{s_k^{(n)}} - 1$

or ce rapport est borné,  $\exists A$  et  $B$  finis tels que

$$\forall n \geq N \quad A < \frac{s_k^{(n+1)}}{s_k^{(n)}} - 1 < B$$

$$\text{or } S = S_0 + \sum_{i=0}^{\infty} \Delta S_i \quad \text{donc } \sum_{i=0}^{\infty} r_1^{(i)} \text{ converge}$$

la remarque précédente entraîne que  $\sum_{i=0}^{\infty} r_k^{(i)}$  converge quel que soit  $k$ , de plus c'est une série à termes positifs donc

$$\lim_{n \rightarrow \infty} n \left( 1 - \frac{r_k^{(n+1)}}{r_{k+1}^{(n)}} \right) \neq 0$$

or  $q_k^{(n)}$  est monotone convergente donc  $\lim_{n \rightarrow \infty} n \Delta q_k^{(n)} = 0$

$$\implies \lim_{n \rightarrow \infty} \frac{\Delta q_k^{(n)}}{1 - \frac{r_{k+1}^{(n+1)}}{r_{k+1}^{(n)}}} = \lim_{n \rightarrow \infty} \frac{n \cdot \Delta q_k^{(n)}}{n \left( 1 - \frac{r_{k+1}^{(n+1)}}{r_{k+1}^{(n)}} \right)} = 0$$

$$\text{donc } \lim_{n \rightarrow \infty} q_{k+1}^{(n)} = \lim_{n \rightarrow \infty} q_k^{(n)}.$$

C.Q.F.D.

Ce théorème est très important : il nous donne une condition de convergence pour les différentes colonnes de  $\epsilon$ -algorithme.

## CHAPITRE V

## APPLICATIONS ET RÉSULTATS NUMÉRIQUES

Ce chapitre est consacré aux applications et aux résultats numériques des algorithmes étudiés. Bien que les conditions des théorèmes énoncés soient parfois difficiles à vérifier nous donnons des exemples pour lesquels ces conditions sont vérifiées. Nous appliquons ces algorithmes à certaines méthodes itératives d'analyse numérique : intégration numérique, résolution de systèmes d'équations, recherches de valeurs propres d'une matrice.

Les résultats numériques seront présentés dans deux formes différentes. La première correspond plus précisément à l'application à posteriori des algorithmes. Elle consiste à donner le tableau triangulaire obtenu :

$$\begin{array}{cccc}
 \theta_0^{(0)} & & & \\
 \theta_0^{(1)} & \theta_2^{(0)} & & \\
 \theta_0^{(2)} & \theta_2^{(1)} & \theta_4^{(0)} & \\
 \theta_0^{(3)} & \theta_2^{(2)} & & \\
 \theta_0^{(4)} & & & 
 \end{array}$$

La seconde correspond mieux à l'application en parallèle des algorithmes. Elle consiste à ne donner au fur et à mesure que le terme d'indice inférieur pair et d'indice supérieur maximum que l'on peut atteindre dans le tableau

triangulaire à partir des valeurs initiales  $\theta_o^{(n)}$

$$\begin{array}{cc}
 \theta_o^{(0)} & \theta_o^{(0)} \\
 \theta_o^{(1)} & \theta_o^{(1)} \\
 \theta_o^{(2)} & \theta_2^{(0)} \\
 \theta_o^{(3)} & \theta_2^{(1)} \\
 \vdots & \vdots \\
 \theta_o^{(2n)} & \theta_{2n}^{(0)} \\
 \theta_o^{(2n+1)} & \theta_{2n}^{(1)}
 \end{array}$$

Les résultats seront systématiquement présentés sous l'une des deux formes. Avant d'étudier les améliorations et les solutions que ces algorithmes peuvent apporter à certains domaines de l'analyse numérique, nous allons donner des exemples simples d'application à des suites et à des séries afin de tester les possibilités respectives des algorithmes.

## V.1 LIMITES DE SUITES ET SOMMES DE SÉRIES.

Ce premier exemple montre qu'il faut prendre certaines précautions pour l'application des algorithmes de type  $\theta$ .

Soit la suite  $S_n = S_{n-1} - 0,005 * S_{n-1}^2$  avec  $S_0 = 0,938$  cette suite est à convergence logarithmique. Nous lui appliquons la première généralisation de l'algèbre avec la suite auxiliaire  $x_n = \text{Log}(1+n)$ . Nous obtenons :

n	$S_n$	$\frac{(n-1)}{8}$
24	0,846	0,9443
25	0,842	0,94440
26	0,839	0,94441
27	0,835	0,94442
28	0,832	0,94443
29	0,828	0,94443
30	0,825	0,944444

On pourrait croire que la limite est de la forme 0,944444... or la limite de la suite initiale est 0 (cependant il ne peut être considéré comme un contre-exemple)

Il existe cependant un certain nombre d'exemples pour lesquels ces procédés d'accélération rendent de grands services. En particulier l'algorithme est très utile pour les suites totalement monotones où totalement oscillante.

Considérons la série :

$$\text{Log}(1+x) = x - \frac{x^2}{2} + \frac{x^3}{3} - \dots$$

Nous savons que cette série converge pour  $x \in ]-1,1]$ , pour  $x=1$  on a  $\text{Log} 2 = 0,6931471805599453$

$S_{2k}$	$\epsilon_{2k}^{(o)}$
1	1
0,83	0,7
0,783	0,6933
0,759	0,69315
0,745	0,6931473
0,736	0,693147185
0,730	0,6931471806
0,725	0,693147180563
0,722	0,693147180560
0,719	0,69314718055994
0,716	0,6931471805599454

pour  $x=2$  bien que la série diverge, l' $\epsilon$ -algorithme converge vers le prolongement analytique de la série, c'est-à-dire vers  $\text{Log}3=1,098612288668110$

$S_{2k}$	$\epsilon_{2k}^{(o)}$
2	2
2,6	1,14
5,06	1,101
12,6	1,0988
37,5	1,09862
121,3	1,098613
410,1	1,0986123
1424,4	1,09861229
5038,5	1,098612289
18069,1	1,09861228869
65504,6	1,098612288669

L' $\epsilon$ -algorithme semble également bien adapté à la sommation des séries trigonométriques.

Prenons par exemple la série :

$$x=2 \left( \sin x - \frac{\sin 2x}{2} + \dots + (-1)^{n+1} \frac{\sin nx}{n} + \dots \right) \quad x \in ]-\pi; \pi[$$

pour  $x = 0,2$  on obtient :

n	$S_n$	$\epsilon_{(n-1)}^{(0)}$
1	0,397	0,397
3	0,384	0,1993
5	0,362	0,20002
7	0,333	0,199999995
9	0,299	0,20000000001
11	0,264	0,1999999999997
13	0,231	0,20000000000000007

Prenons maintenant la série :

$$e^x + \frac{1}{(1-x)^3} = \sum_{k=0}^{\infty} \left( \frac{1}{k!} + a_k \right) x^k \quad \text{avec } a_0=1 \text{ et } a_k = a_{k-1} + k + 1.$$

Cette série possède un pôle d'ordre 3 en  $x=1$  en appliquant le théorème de Montessus de Bollaïre [3] la suite  $\{\epsilon_6^{(n)}\}$  doit converger vers  $S = e^x + \frac{1}{(1-x)^3}$   $\forall x \neq 1$ .

Ainsi pour  $x=1,5$ ,  $S = -3,51831092966193527$ .

Les résultats sont donnés à la page suivante :



$s_n$	$\epsilon_2^{(n)}$	$\epsilon_4^{(n)}$	$\epsilon_6^{(n)}$
- 2			
- 8	- 2,1		
- 22,6	- 2,8	- 1,8	
- 56,9	- 5,5	- 0,9	- 2,5
- 133,0	- 12,6	3,3	- 70,9
- 292,6	- 26,5	9,8	45,2
- 611,5	- 50,9	16,3	90,4
- 1226,6	- 91,4	22,8	- 41,5
- 2379,9	- 157,2	30,5	- 9,8
- 4494,3	- 263,0	40,6	- 4,8
- 8300,2	- 430,9	54,1	- 3,7
- 15047,7	- 695,4	72,4	- 3,56
- 26853,9	- 1109,7	97,2	- 3,52
- 47289,0	- 1755,0	131,2	- 3,519
- 82320,5	- 2755,9	177,6	- 3,5184
- 141874,1	- 4302,8	241,5	- 3,51832
- 242370,7	- 6686,1	329,5	- 3,518312
- 410850,4	- 10348,7	451,2	
- 691649,8	- 15964,7		
- 1157185,8			

## V.2 APPLICATIONS AUX PROBLÈMES D'ANALYSE NUMÉRIQUE.

Pour ces applications nous utiliserons le théorème suivant :

soit une suite  $S_j = S + \sum_{i=1}^p b_i \beta_i^j$  avec  $|\beta_1| > |\beta_2| > \dots > |\beta_p|$  alors

$$\text{alors } \epsilon_{2k}^{(j)} = S + \frac{b_{k+1} \{(\beta_{k+1} - \beta_1) \dots (\beta_{k+1} - \beta_k)\}^2}{\{(1 - \beta_1) \dots (1 - \beta_k)\}^2} \beta_{k+1}^j$$

Dans ce cas nous pouvons voir que si pour  $k \leq p-1 \forall i \beta_i \neq 1$  alors les

limites des rapports  $\frac{\epsilon_{2k}^{(n+1)} - S}{\epsilon_{2k}^{(n)} - S}$  existent pour  $k \leq p-1$  et, nous pouvons appli-

quer les théorèmes du chapitre III

### a) Recherche des valeurs propres d'une matrice.

Brézinski a exposé dans [15] différentes méthodes pour évaluer les éléments propres d'une matrice. Ces méthodes font appel à l' $\epsilon$ -algorithme vectoriel. Soit  $A$  une matrice  $p$  lignes,  $p$  colonnes, on considère  $\{x_n\}$  les itérations de la forme  $x_{n+1} = Ax_n$ . L'application de l' $\epsilon$ -algorithme vectoriel nous donne les éléments  $\epsilon_k^{(n)}$  de  $\mathbb{R}^p$  et nous définissons les quantités :

$$a_k^{(n)} = \frac{(y, \epsilon_{2k}^{(n+1)})}{(y, \epsilon_{2k}^{(n)})} \quad k=0, \dots, p-1 \text{ et } n \in \mathbb{N}$$

où  $y \in \mathbb{R}^p$ ,  $y \neq 0$  et  $(y, \epsilon_{2k}^{(n)})$  est le produit vectoriel. Alors si les valeurs propres vérifient  $|\lambda_1| > |\lambda_2| > \dots > |\lambda_p|$  :

$$\lim_{n \rightarrow \infty} a_k^{(n)} = \lambda_{k+1} \text{ pour } k=0, \dots, p-1$$

de plus :

$$\lim_{n \rightarrow \infty} b_k^{(n)} = \lim_{n \rightarrow \infty} \frac{(y, \varepsilon_{2k+1}^{(n)})}{(n+1)} = \lambda_{k+1} \text{ pour } k=0, \dots, p-1$$

et

$$\lim_{n \rightarrow \infty} \frac{\varepsilon_{2k}^{(n)}}{(y, \varepsilon_{2k}^{(n)})} = \lim_{n \rightarrow \infty} (y, \varepsilon_{2k+1}^{(n)}) \varepsilon_{2k}^{(n)} = v_{k+1} \text{ pour } k=0, \dots, p-1$$

où  $v_{k+1}$  est le vecteur propre associé à  $\lambda_{k+1}$ .

Dans ce cas nous avons :

$$a_k^{(n)} \sim \lambda_{k+1} + \sum_{i=k+2}^p \alpha_i \left( \frac{\lambda_i}{\lambda_{k+1}} \right)^{n+k} + o_{n,k} \text{ ou } o_{n,k} \text{ est négligeable par rapport à la somme.}$$

Si, pour  $k$  fixé, nous appliquons l' $\varepsilon$ -algorithme scalaire aux  $a_k^{(n)}$  le théorème ci-dessus est vérifié et nous pouvons dire que chaque colonne du tableau  $\varepsilon$  obtenu, converge vers  $\lambda_{k+1}$  et cela plus vite que la colonne précédente. Pour cela il suffit que  $\{\lambda_j = \lambda_i \iff i=j\}$

Exemple : soit la matrice

$$\begin{pmatrix} 3 & 12 & 30 \\ -6 & -27 & -66 \\ 4 & 16 & 37 \end{pmatrix} \quad \lambda_1 = 9, \lambda_2 = 3, \lambda_3 = 1$$

nous appliquons l' $\varepsilon$ -algorithme à la suite scalaire  $a_1^{(n)}$

$$\{a_1^{(n)}\} \quad \{\varepsilon_2^{(n)}\} \quad \{\varepsilon_4^{(n)}\} \quad \{\varepsilon_6^{(n)}\}$$

2,93			
2,98	2,99992		
2,992	2,999991	2,999999988	
2,997	2,9999990	2,9999999996	3,0000000002
2,9991	2,99999989	3,0000000001	2,9999999998
2,9997	2,99999998	2,9999999996	
2,99991	2,999999998		
2,99997			

b) Système d'équations.

Soit à résoudre le système d'équations  $Ax = b$  à l'aide d'une technique itérative que nous pouvons exprimer sous la forme :

$$x_{r+1} = P x_r + c$$

où  $P$  est une matrice itérative et  $c$  un vecteur constant

$$x_r - x = P (x_{r-1} - x) = \dots = P^r (x_0 - x)$$

Si les valeurs propres  $\lambda_i$  sont différentes et si les  $x_i$  sont les vecteurs propres correspondants :

$$x_r - x = \alpha_1 \lambda_1^r x_1 + \dots + \alpha_n \lambda_n^r x_n$$

Dans ce cas si on applique l' $\epsilon$ -algorithme scalaire à chaque composante de  $x_r$  les conditions du théorème sont vérifiées si les valeurs propres sont toutes différentes; chaque colonne du tableau  $\epsilon$  "converge plus vite" que la précédente : de plus si  $|\lambda_1| > \dots > |\lambda_i| \gg 1 > |\lambda_{i+1}| > \dots > |\lambda_p|$  les colonnes  $\epsilon_{2j}^{(n)}$  divergent pour  $j \leq i-1$  et convergent pour  $i-1 < j$ .

Exemple : soit à résoudre

$$x_1 + 2x_2 + 10x_3 = 59$$

$$2x_1 + 8x_2 + x_3 = -4$$

$$20x_1 - x_2 + 2x_3 = 74$$

les itérations de Jacobi nous donnent :

$$x_1^{(r+1)} = 59 - 2x_2^{(r)} - 10x_3^{(r)}$$

$$x_2^{(r+1)} = -0,5 - 0,25 x_1^{(r)} - 0,125 x_3^{(r)}$$

$$x_3^{(r+1)} = 37 - 10x_1^{(r)} + 0,5x_2^{(r)}$$

les valeurs propres de la matrice itérative sont :

$$\lambda_1 = 10,027984$$

$$\lambda_2 = -10,015428$$

$$\lambda_3 = 0,012446$$

Les itérations de Jacobi divergent

$x_1^{(n)}$	$\{\epsilon_2^{(n)}\}$	$\epsilon_4^{(n)}$	$\epsilon_6^{(n)}$
0,0			
59	8,1		
- 310	37,4	3,11	
5631,25	509,4	3,001	3,000000000001
- 31503,9	3457,2	3,00001	3,000000000001
565681,6	50775,6	3,0000002	
- 3171510,3	346643		
56854732			

$\{x_2^{(n)}\}$	$\{\epsilon_2^{(n)}\}$	$\{\epsilon_4^{(n)}\}$	$\epsilon_6^{(n)}$
0			
- 0,5	0,13		
- 19,5	2,5	2,04	
146,1	6,8	- 1,97	- 2,000000000000004
- 1799,1	56,8	- 1,9996	- 1,99999999999998
14900,7	485,4	- 1,99995	
- 180692,9	- 5499,3		
1499043,1			

$\{x_3^{(n)}\}$	$\{\epsilon_2^{(n)}\}$	$\{\epsilon_4^{(n)}\}$	$\{\epsilon_6^{(n)}\}$
0			
37			
- 553,2	2,1		
3127	- 44,5	5,5	
- 56202,4	- 338	5,99	
314176,7	- 5064	5,9999	5,9999999999996
- 5649328,7	- 34544	5,999999	6,0000000000002
31624793	- 508333		

c) Problèmes de quadrature numérique.

Genz [24, 25] a généralisé le théorème de ce chapitre au cas où  $S_j = S + \sum_{i=1}^n b_i^{(j)} p_i^{(j)}$  ou  $b_i^{(j)} = b_i^{(j+k_0)}$  pour  $k_0$  entier fixé.

Considérons des intégrales de la forme

$$I = \int_0^1 f(x) dx$$

ainsi qu'une méthode de quadrature approchée :

$$\bar{I} = \sum_{i=1}^n w_i f(x_i) \text{ avec } \sum_{i=1}^n w_i = 1$$

A l'aide de cette formule il est classique de construire des méthodes composites d'intégration de la façon suivante : on écrit que,

$$I = \int_0^{x_1} f(x) dx + \int_{x_1}^{x_2} f(x) dx + \dots + \int_{x_{n-1}}^1 f(x) dx$$

puis chacune de ces intégrales est calculer de façon approchée en utilisant la formule de  $\bar{I}$  correctement modifiée.

Nous appellerons  $I_m$  la valeur approchée de  $I$  obtenue à l'aide d'une méthode composite comprenant  $m$  sous intervalles. On a :

$$I_m = h \sum_{k=0}^{m-1} \sum_{j=1}^n w_j f((x_j+k)h) \text{ avec } h = \frac{1}{m}$$

On démontre que ([69]) sous des conditions faibles sur  $f$  :

$$\lim_{m \rightarrow \infty} I_m = I$$

De plus si  $f$  est analytique sur  $[0, 1]$  alors une simple généralisation de la formule d'Euler-Mac Laurin nous donne [70]

$$I_m = I + a_1 h + a_2 h^2 + \dots + a_k h^k + O(h^{k+1})$$

Si l'on prend  $S_n = I_{2^n}$  ; on aura alors [24]

$$S_n = I + a_1 \lambda_1^n + a_2 \lambda_2^n + \dots + a_k \lambda_k^n + O(\lambda_{k+1}^n)$$

avec  $\lambda_i = 2^{-i}$ .

On peut appliquer l' $\epsilon$ -algorithme à  $S_n$  [71, 72, 73]. Le théorème du début de ce chapitre nous assure la convergence et l'accélération de la convergence à chaque étape. On s'aperçoit dans la pratique [74] que la méthode de Romberg fournit de meilleurs résultats que l' $\epsilon$ -algorithme lorsque  $f$  est continue.

Venons en maintenant au cas où  $f$  présente, à l'une des bornes de l'intervalle d'intégration, une singularité de la forme

$$f(x) = x^\alpha (1-x)^\beta g(x)$$

avec  $g$  analytique dans  $[0,1]$  et  $\alpha, \beta$  non entiers.

Alors on sait que [70]

$$\begin{aligned} I_m = I &+ a_1 h^{1+\alpha} + a_2 h^{2+\alpha} + \dots + a_k h^{k+\alpha} + O(h^{k+\alpha+1}) \\ &+ b_1 h^{1+\beta} + b_2 h^{2+\beta} + \dots + b_k h^{k+\beta} + O(h^{k+\beta+1}) \end{aligned}$$

Si l'on applique la méthode de Romberg il faut traiter séparément les termes en  $\alpha$  et ceux en  $\beta$  [75].

Par contre si nous considérons la suite  $\{S_n = I_{2^n}\}$  alors :

$$S_n = I + a_1 \lambda_1^n + a_2 \lambda_2^{(n)} + \dots + b_1 \delta_1^{(n)} + \dots + b_k \delta_k^{(n)}$$

avec  $\lambda_i = 2^{-(\alpha+i)}$   $\delta_i = 2^{-(\beta+i)}$ .

L'application de l' $\epsilon$ -algorithme sera performant, la convergence et l'accélération de convergence sont assurées. Pour des singularités logarithmiques on obtient des développements limités analogues et l' $\epsilon$ -algorithme est également très efficace [76].

Si maintenant  $f$  possède une singularité au milieu de l'intervalle, par exemple :

$$f(x) = (x-a)^\alpha g(x) \quad 0 < a < 1$$

et  $g$  analytique sur  $[0,1]$  alors [70]

$$I_m = I + a_1 h + \dots + a_k h^k + O(h^{k+1}) \\ + b_1^{(m)} h^{\alpha+1} + \dots + b_k^{(m)} h^{\alpha+k} + O(h^{k+\alpha+1})$$

avec cette fois les coefficients  $b_i^{(m)}$  qui dépendent de  $m$  de sorte que :

$$b_i^{(m)} = y_i(m\alpha) \text{ et } y_i(m\alpha) = y_i(m\alpha+1)$$

Si  $\alpha$  est un rationnel il aura un développement binaire périodique ; soit  $p$  cette période alors :

$$b_i^{(2^n)} = b_i^{(2^{n+p})}$$

Par conséquent la suite  $\{S_n = I_{2^n}\}$  vérifiera

$$S_n = I + \sum_{i=1}^k a_i \lambda_i^n + \sum_{i=1}^k c_i^{(n)} \delta_i^n$$

avec  $\lambda_i = 2^{-i}$ ,  $c_i^{(n)} = b_i^{(2^n)}$  et  $\delta_i = 2^{-(\alpha+i)}$ .

Là encore l'application de l' $\epsilon$ -algorithme est efficace de plus Genz à démontrer que  $\epsilon_{2kp}^{(n)} = I$ .

Exemples :

$$1) \quad I = \int_0^1 x^{0,75} \cos x \, dx = 0,44516493$$

en utilisant une formule de quadrature de Gauss-Legendre avec deux points et en appliquant l' $\epsilon$ -algorithme nous obtenons :

$\{\epsilon_0^{(n)}\}$	$\{\epsilon_2^{(n)}\}$	$\{\epsilon_4^{(n)}\}$
0,447		
0,4459	0,44501	
0,4454	0,445158	0,44516491
0,4452	0,445164	
0,44518		

$$2) \quad I = \int_0^1 \frac{dx}{x^2 - 0,01} = -1,0033534$$

en utilisant une formule de quadrature de Gauss-Legendre avec huit points :

$\epsilon_0^{(n)}$	$\epsilon_2^{(n)}$	$\epsilon_4^{(n)}$	
331,0			
15,252	86,6		
107,535	54,4	155,7	
- 107,2	- 371,4	- 1,0033535	- 1,0033534
- 109,5	- 56,4	- 1,0033534	
15,2	369,4		
107,5			

Nous pouvons également appliquer l' $\epsilon$ -algorithme à la recherche de zéros multiple d'un polynôme. Il est aussi très efficace pour accélérer les suites totalement monotones non logarithmique et les suites totalement oscillantes. On montre que les sommes partielles de certaines séries d'opérateurs utilisées en analyse numérique sont totalement monotones. C'est le cas de la série d'interpolation de Newton, la série de Newton pour la dérivation, de la série d'intégration de Newton-Gregory, la formule d'Euler-Mac Laurin pour certaines fonctions ... Wynn [77] a donné des exemples qui montrent le gain appréciable que procure l' $\epsilon$ -algorithme appliqué à ce type de suite.

## CONCLUSION

Nous voyons que les méthodes d'accélération de la convergence sont intéressantes. Il semble indiscutable que leur emploi dans les méthodes itératives ainsi que les diverses applications que nous avons examinées sont d'un grand intérêt pratique.

Il est cependant essentiel si l'on veut espérer obtenir de bons résultats numériques avec des algorithmes d'accélération de la convergence de les programmer de façon extrêmement soignée. La propagation des erreurs dues à l'ordinateur est quelques fois catastrophique. Il est donc nécessaire d'utiliser les règles particulières et même parfois de corriger l'arithmétique de l'ordinateur. On trouvera dans [4, 78, 79] des programmes FORTRAN et dans [80, 81] des programmes algol.

La question la plus importante qui reste à résoudre est celle du choix du procédé d'accélération de la convergence le mieux adapté à la suite à accélérer.

Dans cette thèse nous avons essayé de répondre, en partie, à cette question :

- si la suite est à convergence linéaire mais non logarithmique l' $\epsilon$ -algorithme et le  $\theta$ -algorithme sont bien adaptés

- si la suite est à convergence logarithmique l' $\epsilon$ -algorithme nous donne de mauvais résultats tandis que le  $\theta$ -algorithme de Brezinski et parfois le  $\rho$ -algorithme sont intéressants

- l' $\epsilon$ -algorithme est également bien adapté aux suites totalement monotones non logarithmique et aux suites totalement oscillantes.

De nombreuses questions restent à aborder :

- soit  $\{S_n\}$  une suite totalement monotone (resp. totalement oscillante), la question est de savoir si  $\{\varepsilon_{2k}^{(n)}\}$  est totalement monotone (resp. totalement oscillante). Dans ce cas il y aurait accélération de la convergence à chaque étape

- extension des algorithmes de type  $\theta$  au cas non scalaire
- peut-on changer une suite monotone en une suite oscillante par laquelle l' $\varepsilon$ -algorithme est plus stable (ceci a été abordé par DANIEL mais sa méthode nécessite beaucoup de termes de la suite initiale).
- le champ des applications reste à élargir.

Enfin nous avons, dans cette exposé, abordé ces algorithmes de façon très technique, je pense qu'une approche plus théorique nous fournirait des résultats plus généraux et plus complets. Mais le plus difficile est finalement de convaincre ceux qui utilisent les méthodes itératives d'employer systématiquement des méthodes d'accélération de la convergence.

## RÉFÉRENCES

- [1] AITKEN (A.C.). "On Bernoulli's numerical solution of algebraic equations". Proc. Roy. Soc. Edinburgh, vol. 46 p. 289-305 (1926).
- [2] BAUER (F.L.). "The  $g$ -algorithm". J. Soc. Indust. Appl. Math. vol. 8 n°1 march 1960.
- [3] BREZINSKI (C.). "Accélération de la convergence en analyse numérique". Cours de DEA, Lille.
- [4] BREZINSKI (C.). "Méthode d'accélération de la convergence en analyse numérique". Thèse, Grenoble 1971.
- [5] BREZINSKI (C.). "Résultats sur les procédés de sommation et l' $\epsilon$ -algorithme". RIRO, R3 (1970) p. 147-153.
- [6] BREZINSKI (C.). "Conditions d'application et de convergence de procédés d'extrapolation". Numer. Math., 20 (1972) p. 64-79.
- [7] BREZINSKI (C.). "Généralisation des extrapolations polynomiales et rationnelles". RAIRO, R1 (1972) p. 61-66.
- [8] BREZINSKI (C.), CROUZEIX (M.). "Remarques sur le procédé  $\Delta^2$  d'Aitken". CRAS, 270 (1970) p. 896-898.
- [9] BREZINSKI (C.). "Etudes sur les  $\epsilon$  et  $\rho$ -algorithmes". Numer. Math., 17 (1971) p. 153-162.
- [10] BREZINSKI (C.). "L' $\epsilon$ -algorithme et les suites totalement monotones et oscillantes". C.R. Acad. Sc. Paris 276A (1973) p. 305-308.
- [11] BREZINSKI (C.). "Accélération de suites à convergence logarithmique". C.R. Acad. Sc. Paris 273A (1971) p. 727-730.
- [12] BREZINSKI (C.). "Comparaison de suites convergentes". RIRO, R2, 1971, p. 95-99.
- [13] BREZINSKI (C.). "Notions sur la comparaison de suites convergentes". Séminaire d'analyse numérique Lille 14 mars 1973.
- [14] BREZINSKI (C.). "Application du  $\rho$ -algorithme à la quadrature numérique". C.R. Acad. Sc. Paris, 270A (1970) p. 1252-1253.
- [15] BREZINSKI (C.). "Computation of the Eigenlements of a matrix by the  $\epsilon$ -algorithm". Linear Algebra and its applications 11, p. 7-20 (1975).
- [16] BURLIRSCH (R.), STOER (J.). "Fehlerabschätzungen und extrapolation mit rationalen funktionen bei verfahren von Richardson typus". Numer. Math., 6 (1964) p. 413-427.

- [17] CLARK (W.D.). "Infinite series transformations and their applications". Doctoral Thesis, Austin Texas, 1967.
- [18] CLARK (W.D.), GRAY (H.L.), ADAMS (J.E.). "A note on the T transformation of Lubkin". J. Research NBS vol. 73B n°1 p. 25-29 1969.
- [19] CORDELLIER (F.). "Caractérisation des suites que la première étape du  $\theta$ -algorithme transforme en suites constantes". (à paraître).
- [20] CORDELLIER (F.). "Interprétation géométrique d'une étape de l' $\epsilon$ -algorithme". Séminaire d'analyse numérique, Lille, 1973.
- [21] CORDELLIER (F.). "Mise en oeuvre d'une méthode stable de l' $\epsilon$ -algorithme". Séminaire d'analyse numérique, Lille, 1975.
- [22] CORDELLIER (F.). "Séminaire d'analyse numérique de Lille I". 1975.
- [23] DANIEL (J.W.). "Summation of series of positive terms by condensation transformation". Math. Comp. vol. 23 n°105 p. 91-96, 1969.
- [24] GENZ (A.). "The  $\epsilon$ -algorithm and some other applications of Padé approximants in numerical analysis". in "Padé approximants", P.R. Graves-Morris ed., the institute of physics, London, 1973.
- [25] GENZ (A.). "Applications of the  $\epsilon$ -algorithm to quadrature problems". in "Padé approximants and their applications", P.R. Graves-Morris ed., Academic Press, New York, 1973.
- [26] GERMAIN-BONNE (B.). "Accélération de la convergence d'une suite par extrapolation". Colloque d'analyse numérique d'Anglet, juin 1971.
- [27] GERMAIN-BONNE (B.). "Transformations non linéaires de suites". Séminaire d'analyse numérique, Lille, 28 mars 1973.
- [28] P.R. GRAVES-MORRIS ed. "Padé approximants". The institute of physics, London and Bristol, 1973.
- [29] GRAY (H.L.), ATCHISON (T.A.). "A note on the G-transformation". Journal of Research NBS vol. 72 n°1, 1968.
- [30] GRAY (H.L.), ATCHISON (T.A.). "Applications of the G and B transforms to the Laplace transform". Proceedings ACM National Conference, 1968.
- [31] GRAY (H.L.), ATCHISON (T.A.). "Nonlinear transformations related to the evaluations of improper integrals I". SIAM J. Numer. Anal. vol.4 n°3 part.1, 1967.
- [32] GRAY (H.L.), ATCHISON (T.A.). "Nonlinear transformations related to the evaluations of improper integrals II". SIAM J. Numer. Anal. vol.5 part.2, 1968.
- [33] GRAY (H.L.), ATCHISON (T.A.). "The generalized G-transform". Math. of Comp. vol.22, n°103, p. 595-606, 1968.
- [34] GRAY (H.L.), CLARK (W.O.). "On a class of nonlinear transformations and their applications to the evaluation of infinite series". J. Res. NBS vol. 73B n°0 p. 251-274, 1969.

- [35] GRAY (H.L.), SCHUCANY (W.R.). "Some limiting cases of the G transformation". Math. of Comp. vol. 23 n°108 October 1969.
- [36] GRAY (H.L.), ATCHISON (T.A.), MC WILLIAMS (G.V.). "Higher order G transformations". SIAM J. Numer. Anal., 8 (1971) p. 365-381.
- [37] LARKIN (F.M.). "Some techniques for rational interpolation". Computer J., 10 (1967), p. 178-187.
- [38] LAURENT (P.J.). "Etude des procédés d'extrapolation en analyse numérique". Thèse, Grenoble, 1964.
- [39] LUBKIN (S.). "A method of summing infinite series". J. Res. NBS vol.48 n°3, p. 229-254, mars 1952.
- [40] MIKELADZE (S.E.). "Numerical methods of mathematical analysis". AEC-TR 4285.
- [41] MILNE-THOMSON. "The calculus of finite differences". Macmillan, London, 1965.
- [42] OVERHOLT (K.K.). "Extended Aitken acceleration". B.I.T. 5 (1965) p. 122-132.
- [43] PADE (H.). "Sur la représentation approchée d'une fonction par des fractions rationnelles". Ann. Ec. Norm. Sup., 9 (1892) p. 1-93.
- [44] PETIT (D.). "Mémoire de D.E.A. : Etude de la G transformation et du procédé q". Lille Octobre 1975.
- [45] PERRON (O.). "Die Lehre von dem Kettenbrüchen". Chelsea, New York, 1950.
- [46] FYE (W.C.), ATCHISON (T.A.). "An algorithm for the computation of higher order G-transformation". SIAM J. Numer. Anal., 10 (1973) p. 1-7.
- [47] RICHARDSON (L.F.). "The deferred approach to the limit". Trans. Phil. Roy. Soc. 226 (1927) p. 261-299.
- [48] RUTISHAUSER (H.). "Der Quotienten-Differenzen-Algorithmus". Birkhauser Verlag 1957.
- [49] SCHMIDT (J.R.). "On the numerical solution of linear simultaneous equations by an iterative method". Phil. Mag. ser. 7 vol. 32 p. 369-383, 1951.
- [50] SHANKS (D.). "Nonlinear transformations of divergent and slowly convergent series". J.Math. Phys. vol. 34 n°1, p. 1-42, avril 1955.
- [51] STREIT (P.). "The  $T_m$  transformation". Math. of Computation vol. 30 n° 135, p. 505-511 (1976).
- [52] WALL (H.). "Analytic theory of continued fractions". Van Nostrand, New York, 1948.
- [53] WOINTRAIN (J.). Mémoire de D.E.A., Lille, Octobre 1974.
- [54] WUYTACK (L.). "A new technique for rational extrapolation to the limit". Numer. Math. 17 (1971) p. 215-221.

- [55] WYNN (P.). "On a device for computing the  $e_m(S_n)$  transformation". MTAC vol. 10 n°54 p. 91-96 avril 1956.
- [56] WYNN (P.). "Singular rules for certain nonlinear algorithms". Nordisk. Tidskr. Informations Behandling 3 (1963) p. 175-193.
- [57] WYNN (P.). "On a sufficient condition for the instability to the  $\epsilon$ -Algorithm". Nieuw. Arch. Wisk 9 (1961) p. 117-119.
- [58] WYNN (P.). "On the propagation of error in certain non linear algorithms". Numer. Math. 1 (1959) p. 142-149.
- [59] WYNN (P.). "On the convergence and stability of the epsilon algorithm". J. SIAM Numer. Analy. vol.3, n°1, (1966) p. 91-121.
- [60] WYNN (P.). "L' $\epsilon$ -algorithmo e la tavola di Padé". Rend. di Math. Roma 20 (1961) p.403.
- [61] WYNN (P.). "The rational approximation of functions which are formally defined by a power series expansion". Math. of Comp. vol. 14 n°70 (avril 1960).
- [62] WYNN (P.). "Upon systems of recursions which obtain among the quotients of the Pade table". Numer. Math. vol.8 n°3 (1966).
- [63] WYNN (P.). "On a procustean technique for the numerical transformation of slowly convergent sequences and series". Proc. Cambridge Phil. Soc. 52 (1956).
- [64] WYNN (P.). "Note on programming repeated application of the  $\epsilon$ -algorithm". Chiffres vol.8 n°1 p. 23-62, 1965.
- [65] WYNN (P.). "Hierarchies of arrays and functions sequences associated with the  $\epsilon$ -algorithm and its first confluent form". Centre de recherches mathématiques, Université de Montréal, report CRM (1972).
- [66] BREZINSKI (C.). "Méthodes d'accélération de la convergence en analyse numérique". Lecture notes in Math. Springer-Verlag.
- [67] CABANNES. "Pade Approximants Method".
- [68] ESPINOZA (C.). "Application de  $\epsilon$ -algorithm à des suites non-scalaires, comparaison des  $\epsilon$ ,  $\rho$ ,  $\theta$  algorithmes". Mémoire de DEA, Lille I, 1974.

- [69] DAVIS (P.J.), RABINOWITZ (P.). "Numerical integration" Blaisdell Waltham, 1967.
- [70] LYNES (J.N.), NINHAM (B.W.). "Numerical quadrature and asymptotic expansions" Math. Comp. 21 (1967) 162-178.
- [71] CHISHOLM (J.S.R.). "Application of Padé approximation to numerical integration" Rocky Mountains J. Math. 4 (1974) 159-168.
- [72] CHISHOLM (J.S.R.). "Padé approximation of single variable integrals" Colloquium on computational methods in theoretical physics, Marseille 1970.
- [73] CHISHOLM (J.S.R.). "Accelerated convergence of sequences of quadrature approximants" second colloquium on computational methods in theoretical physics, Marseille 1971
- [74] CHISHOLM (J.S.R.), GENZ (A.C.), ROWLANDS (G.E.). "Accelerated convergence of sequences of quadrature approximation" J. Comp. Phys. 10 (1972)
- [75] FOX (L.). "Romberg integration for a class of singular integrands" Computer J., 10 (1967) 87-93
- [76] KAHANER (D.K.). "Numerical quadrature by  $\epsilon$ -algorithm" Math. Comp. 26 (1972) 689-694
- [77] WYNN (P.). "Transformations de série à l'aide de l' $\epsilon$ -algorithme" C.R.A.S. Paris 275 A (1972) 1351-1353
- [78] BREAUX (L.C.). "A numerical study of the application of acceleration techniques and prediction algorithms to numerical integration" M.Sc thesis, Louisiana State Univ., New OrLéans, 1971
- [79] BREZINSKI (C.). "Méthodes numériques générales pour l'accélération de la convergence" à paraître
- [80] "Procedures Algol en analyse numérique", Tome 2 CNRS, Paris, 1972
- [81] WYNN (P.). "An arsenal of Algol procedures for the evaluation of continued fractions and for effecting the  $\epsilon$ -algorithm", chiffres 9 (1966)