



HAL
open science

Méthodes d'accélération de la convergence en analyse numérique

Claude Brezinski

► **To cite this version:**

Claude Brezinski. Méthodes d'accélération de la convergence en analyse numérique. Modélisation et simulation. Université Joseph-Fourier - Grenoble I, 1971. tel-00282774

HAL Id: tel-00282774

<https://theses.hal.science/tel-00282774>

Submitted on 28 May 2008

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

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

THÈSES

PRÉSENTÉES

A L'UNIVERSITÉ SCIENTIFIQUE ET MÉDICALE DE GRENOBLE

POUR OBTENIR

LE GRADE DE DOCTEUR ÈS SCIENCES MATHÉMATIQUES

PAR

CLAUDE BREZINSKI



PREMIÈRE THÈSE

MÉTHODES D'ACCÉLÉRATION DE LA CONVERGENCE
EN ANALYSE NUMÉRIQUE

DEUXIÈME THÈSE

PROPOSITIONS DONNÉES PAR LA FACULTÉ

ORGANISATION DES FICHIERS D'UNE BANQUE DE DONNÉES

Soutenues, le 26 Avril 1971, devant la Commission d'Examen

MM. J. KUNTZMANN *Président*

N. GASTINEL

P. J. LAURENT *Examineurs*

F. CESCHINO

THÈSES

PRÉSENTÉES

A L'UNIVERSITÉ SCIENTIFIQUE ET MÉDICALE DE GRENOBLE

POUR OBTENIR

LE GRADE DE DOCTEUR ÈS SCIENCES MATHÉMATIQUES

PAR

CLAUDE BREZINSKI



PREMIÈRE THÈSE

MÉTHODES D'ACCÉLÉRATION DE LA CONVERGENCE
EN ANALYSE NUMÉRIQUE

DEUXIÈME THÈSE

PROPOSITIONS DONNÉES PAR LA FACULTÉ

ORGANISATION DES FICHIERS D'UNE BANQUE DE DONNÉES

Soutenues, le 26 Avril 1971, devant la Commission d'Examen

MM. J. KUNTZMANN *Président*

N. GASTINEL

P. J. LAURENT *Examineurs*

F. CESCHINO

PROFESSEURS TITULAIRES

Président : Monsieur Michel SOUTIF
Vice-Président : Monsieur Gabriel CAU

MM.	ANGLES D'AURIAC Paul	Mécanique des fluides
	ARNAUD Georges	Clinique des maladies infectieuses
	ARNAUD Paul	Chimie
	AYANT Yves	Physique approfondie
	BARBIER Jean-Claude	Physique expérimentale
	BARBIER Reynold	Géologie appliquée
	BARJON Robert	Physique nucléaire
	BARNOUD Fernand	Biosynthèse de la cellulose
	BARRIE Joseph	Clinique chirurgicale
	BENOIT Jean	Radioélectricité
	BESSON Jean	Electrochimie
	BEZES Henri	Chirurgie générale
	BLAMBERT Maurice	Mathématiques pures
	BOLLIET Louis	Informatique (IUT B)
	BONNET Georges	Electrotechnique
	BONNET Jean-Louis	Clinique ophtalmologique
	BONNET-EYMARD Joseph	Pathologie médicale
	BONNIER Etienne	Electrochimie Electrometallurgie
	BOUCHERLE André	Chimie et Toxicologie
	BOUCHEZ Robert	Physique nucléaire
	BRAVARD Yves	Géographie
	BRISSONNEAU Pierre	Physique du Solide
	BUYLE-BODIN Maurice	Electronique
	CABANAC Jean	Pathologie chirurgicale
	CABANEL Guy	Clinique rhumatologique et hydrologique
	CALAS François	Anatomie
	CARRAZ Gilbert	Biologie animale et pharmacodynamie
	CAU Gabriel	Médecine légale et Toxicologie
	CAUQUIS Georges	Chimie organique
	CHABAUTY Claude	Mathématiques pures
	CHATEAU Robert	Thérapeutique
	CHENE Marcel	Chimie papetière
	COEUR André	Pharmacie chimique
	CONTAMIN Robert	Clinique gynécologique
	COUDERC Pierre	Anatomie Pathologique
	CRAYA Antoine	Mécanique
Mme	DEBELMAS Anne-Marie	Matière médicale
MM.	DEBELMAS Jacques	Géologie générale
	DEGRANGE Charles	Zoologie
	DESSAUX Georges	Physiologie animale
	DODU Jacques	Mécanique appliquée
	DREYFUS Bernard	Thermodynamique
	DUCROS Pierre	Cristallographie
	DUGOIS Pierre	Clinique de Dermatologie et Syphiligraphie
	FAU René	Clinique neuro-psychiatrique
	FELICI Noël	Electrostatique
	GAGNAIRE Didier	Chimie physique
	GALLISSOT François	Mathématiques pures
	GALVANI Octave	Mathématiques pures
	GASTINEL Noël	Analyse numérique
	GERBER Robert	Mathématiques pures

MM.	GIRAUD Pierre	Géologie
	KLEIN Joseph	Mathématiques pures
Mme	KOFLER Lucie	Botanique et physiologie végétale
MM.	KOSZUL Jean-Louis	Mathématiques pures
	KRAVTCHENKO Julien	Mécanique
	KUNTZMANN Jean	Mathématiques appliquées
	LACAZE Albert	Thermodynamique
	LACHARME Jean	Biologie végétale
	LATURAZE Jean	Biochimie pharmaceutique
	LEDRU Jean	Clinique médicale B
	LLIBOUTRY Louis	Géophysique
	LOUP Jean	Géographie
Mlle	LUTZ Elisabeth	Mathématiques pures
	MALGRANGE Bernard	Mathématiques pures
	MALINAS Yves	Clinique obstétricale
	MARTIN-NOEL Pierre	Seméiologie médicale
	MASSEPORT Jean	Géographie
	MAZARE Yves	Clinique médicale A
	MICHEL Robert	Minéralogie et Pétrographie
	MOURIQUAND Claude	Histologie
	MOUSSA André	Chimie nucléaire
	NEEL Louis	Physique du Solide
	OZENDA Paul	Botanique
	PAUTHNET René	Electrotechnique
	PAYAN Jean-Jacques	Mathématiques pures
	PEBAY-PEYROULA Jean-Claude	Physique
	PERRET René	Servomécanismes
	PILLET Emile	Physique industrielle
	RASSAT André	Chimie systématique
	RENARD Michel	Thermodynamique
	REULOS René	Physique industrielle
	RINALDI Renaud	Physique
	ROGET Jean	Clinique de pédiatrie et de puériculture
	SANTON Lucien	Mécanique
	SEIGNEURIN Raymond	Microbiologie et Hygiène
	SENGEL Philippe	Zoologie
	SILBERT Robert	Mécanique des fluides
	SOUTIF Michel	Physique générale
	TANCHE Maurice	Physiologie
	TRAYNARD Philippe	Chimie générale
	VAILLAND François	Zoologie
	VAUQUOIS Bernard	Calcul électronique
Mme	VERAIN Alice	Pharmacie galénique
	VERAIN André	Physique
Mme	VEYRET Germaine	Géographie
MM.	VEYRET Paul	Géographie
	VIGNAIS Pierre	Biochimie médicale
	YOCOZ Jean	Physique nucléaire théorique

PROFESSEURS ASSOCIES

MM.	BULLEMER Bernhard	Physique
	RADHAKRISHNA Pidatala	Thermodynamique

PROFESSEURS SANS CHAIRE

MM.	AUBERT Guy	Physique
Mme	BARBIER Marie-Jeanne	Electrochimie
MM.	EARRA Jean	Mathématiques appliquées
	BEAUDOING André	Pédiatrie
	BERTRANDIAS Jean-Paul	Mathématiques appliquées
	BLAREZ Jean-Pierre	Mécanique
	BONNETAIN Lucien	Chimie minérale
Mme	BONNIER Jane	Chimie générale
MM.	CARLIER Georges	Biologie végétale
	COHEN Joseph	Electrotechnique
	COUMES André	Radioélectricité
	DEPASSEL Roger	Mécanique des Fluides
	DEPORTES Charles	Chimie minérale
	DESRE Pierre	Métallurgie
	DOLIQUE Jean-Michel	Physique des plasmas
	GAUTHIER Yves	Sciences biologiques
	GEINORE Michel	Electroradiologie
	GIDON Paul	Géologie et Minéralogie
	GLENAT René	Chimie organique
	HACQUES Gérard	Calcul numérique
	JANIN Bernard	Géographie
Mme	KAHANE Josette	Physique
MA.	LATREILLE René	Chirurgie générale
	LAURENT Pierre	Mathématiques appliquées
	MULLER Jean-Michel	Thérapeutique
	PERRIAUX Jean-Jacques	Géologie et minéralogie
	POULOUJADOFF Michel	Electrotechnique
	REBECQ Jacques	Biologie (CUS)
	REVOL Michel	Urologie
	REYMOND Jean-Charles	Chirurgie générale
	ROBERT André	Chimie papetière
	SARRAZIN Roger	Anatomie et chirurgie
	SARROT-REYNAULD Jean	Géologie
	SIBILLE Robert	Construction Mécanique
	SIROT Louis	Chirurgie générale
Mme	SOUTIF Jeanne	Physique générale
M.	VALENTIN Jacques	Physique nucléaire

MAITRES DE CONFERENCES ET MAITRES DE CONFERENCES AGREGES

Mle	ACNIUS-DELORE Claudine	Physique pharmaceutique
	ALARY Josette	Chimie analytique
MM.	AMBLARD Pierre	Dermatologie
	AMBROISE-THOMAS Pierre	Parasitologie
	ARMAND Yves	Chimie
	BEGUIN Claude	Chimie organique
	BELOKIZKY Elie	Physique
	BENZAKEN Claude	Mathématiques appliquées
	BERTRANDIAS Françoise	Mathématiques pures
	BLIMAN Samuel	Electronique (EIE)
	BLOCH Daniel	Electrotechnique
Mme	POUCHE Liane	Mathématiques (CUS)
MM.	BOUCHET Yves	Anatomie
	BOUSSARD Jean-Claude	Mathématiques appliquées
	BOUVARD Maurice	Mécanique des Fluides
	BRIERE Georges	Physique expérimentale

	BRODEAU François	Mathématiques (IUT B)
	BRUCEL Lucien	Energétique
	EUISSON Roger	Physique
	BUTEL Jean	Orthopédie
	CHAMBAZ Edmond	Biochimie médicale
	CHAMPETIER Jean	Anatomie et organogénèse
	CHARACHON Robert	Oto-Rhino-Laryngologie
	CHIAVERINA Jean	Biologie appliquée (EFP)
	CHIBON Pierre	Biologie animale
	COHEN-ADDAD Jean-Pierre	Spectrométrie physique
	COLOMB Maurice	Biochimie médicale
	CONTE René	Physique
	CROUZET Guy	Radiologie
	DURAND Francis	Métallurgie
	DUSSAUD René	Mathématiques (CUS)
Mme	ETERRADOSSI Jacqueline	Physiologie
MM.	FAURE Jacques	Médecine légale
	GAVEND Michel	Pharmacologie
	GENSAC Pierre	Botanique
	GERMAIN Jean-Pierre	Mécanique
	GIDON Maurice	Géologie
	GRIFFITHS Michael	Mathématiques appliquées
	GROULADE Joseph	Biochimie médicale
	HOLLARD Daniel	Hématologie
	HUGONOT Robert	Hygiène et médecine préventive
	IDELMAN Simon	Physiologie animale
	IVANES Marcel	Electricité
	JALBERT Pierre	Histologie
	JOLY Jean-René	Mathématiques pures
	JOUBERT Jean-Claude	Physique du Solide
	JULLIEN Pierre	Mathématiques pures
	KAHANE André	Physique générale
	KUHN Gérard	Physique
Mme	LAJZEROWICZ Jeannine	Physique
MM.	LAJZEROWICZ Joseph	Physique
	LANCIA Roland	Physique atomique
	LE JENTER Noël	Electronique
	LEROY Philipps	Mathématiques
	LOISEAUX Jean-Marie	Physique nucléaire
	LONGEQUEUE Jean-Pierre	Physique nucléaire
	LUN DUC Cuong	Chimie organique
	MACHE Régis	Physiologie végétale
	MAGNIN Robert	Hygiène et Médecine préventive
	MARSHAL Jean	Mécanique
	MARTIN-BOUYER Michel	Chimie (CUS)
	MAYNARD Roger	Physique du Solide
	MICAUD Max	Maladies infectieuses
	MOREAU René	Hydraulique (INP)
	NEGRE Robert	Mécanique
	PARAMELLE Bernard	Pneumologie
	PECCOUD François	Analyse (IUT B)
	PEFFEN René	Métallurgie
	PELMONT Jean	Physiologie animale
	PERRET Jean	Neurologie
	PERRIN Louis	Pathologie expérimentale
	PFISTER Jean-Claude	Physique du Solide
	PHELIP Xavier	Rhumatologie
Mlle	PIERY Yvette	Biologie animale

MM.	RACHAIL Michel	Médecine interne
	RACINET Claude	Gynécologie et obstétrique
	RICHARD Lucien	Botanique
Mme	RINAUDO Marguerite	Chimie macromoléculaire
MM.	ROMIER Guy	Mathématiques (IUT B)
	ROUGEMONT (DE) Jacques	Neuro-chirurgie
	STIEGLITZ Paul	Anesthésiologie
	STOEBNER Pierre	Anatomie pathologique
	VAN CURSEM Bernard	Mathématiques appliquées
	VEILLON Gérard	Mathématiques appliquées (INP)
	VIALON Pierre	Géologie
	VOOG Robert	Médecine interne
	VROUSSOS Constantin	Radiologie
	ZADWORNÝ François	Electronique

MAITRES DE CONFERENCES ASSOCIES

MM.	BOUDOURIS Georges	Radioélectricité
	CHEEKE John	Thermodynamique
	GOLDSCHMIDT Hubert	Mathématiques
	YACOUD Mahmoud	Médecine légale

CHARGES DE FONCTIONS DE MAITRES DE CONFERENCES

Mme	BERIEL Hélène	Physiologie
Mme	RENAUDET Jacqueline	Microbiologie

Fait le 9 avril 1971

S O M M A I R E

<u>INTRODUCTION</u>	1
<u>CHAPITRE I</u> Les procédés de sommation	9
1.1. Généralités	10
1.2. Principales méthodes	12
1.3. Accélération de la convergence	13
1.4. Interpretation des procédés de sommation	18
<u>CHAPITRE II</u> Les procédés d'extrapolation	23
2.1. Le problème de l'extrapolation	24
2.2. Le procédé d'extrapolation de Richardson	26
2.3. L' ϵ -algorithme	28
2.3.1. Le procédé Δ^2 d'Aitken et l' ϵ -algorithme ..	28
2.3.2. Propriétés de l' ϵ -algorithme	32
2.3.3. Connection avec la table de Padé	39
2.3.4. Connection avec les fractions continues	42
2.3.5. Généralisation aux cas non scalaires	48
2.4. Le ξ -algorithme	49
2.4.1. Forme étendue du ξ -algorithme	53
2.4.2. Le ξ^r -algorithme	53
2.4.3. Propriétés du ξ -algorithme	55
2.5. Les algorithmes de losange	55
2.5.1. La décomposition g	56
2.5.2. Equations aux dérivées partielles associées aux algorithmes de losange	59
2.6. Accélération des suites Hypergéométriques	63
<u>CHAPITRE III</u> Etude de la convergence	71
3.1. Théorèmes généraux de convergence	72
3.2. Théorèmes particuliers de convergence	75
3.3. Accélération de la convergence	82
3.4. Comparaison avec le procédé de Richardson	85
3.5. Liaison avec les développements asymptotiques	87
<u>CHAPITRE IV</u> Formes confluentes de l' ϵ -algorithme	89
4.1. Première forme confluyente de l' ϵ -algorithme	92

4.1.1.	Exactitude de $\mathcal{E}_{2k}(t)$	93
4.1.2.	Convergence de $\mathcal{E}_{2k}(t)$	96
4.1.3.	Introduction d'un facteur d'accélération ..	98
4.1.4.	Introduction d'une fonction d'accélération.	102
4.2.	Seconde forme confluente de l' \mathcal{E} -algorithme	105
4.2.1.	Exactitude de $\mathcal{E}_{2k}^*(t)$	107
4.2.2.	Convergence de $\mathcal{E}_{2k}^*(t)$	108
4.2.3.	Introduction d'un facteur d'accélération ..	109
4.2.4.	Introduction d'une fonction d'accélération.	111
4.3.	Formes confluentes et développements asymptotiques	113
4.4.	Théorèmes particuliers de convergence	116
<u>CHAPITRE V</u>	Programmation des algorithmes	119
5.1.	Considérations théoriques	120
5.1.1.	Règles particulières	120
5.1.2.	Stabilité et propagation des erreurs	123
5.2.	Application des algorithmes	125
<u>CHAPITRE VI</u>	Applications et résultats numériques	129
6.1.	Applications des formes discrètes	130
6.1.1.	Limites de suites et sommes de séries	131
6.1.2.	Méthodes itératives en analyse numérique linéaire	135
6.1.2.1.	Résolution de système linéaire	135
6.1.2.2.	Calcul du rayon spectral d'une matrice	137
6.1.3.	Résolution des équations	139
6.1.4.	Interpolation inverse	150
6.1.5.	Comparaison avec l'extrapolation polynômiale	151
6.1.5.1.	Application à la quadrature numérique	153
6.1.5.2.	Application à la dérivation numérique	157
6.2.	Applications des formes confluentes	158
6.2.1.	Limite d'une fonction	161
6.2.2.	Calcul des intégrales impropres	162
6.2.3.	Résolution des équations	164
6.2.4.	Seconde forme discrète de l' \mathcal{E} -algorithme .	167
ANNEXE	169
BIBLIOGRAPHIE	177

NOTATIONS

E espace vectoriel topologique

E' dual topologique de E c'est à dire l'espace des formes linéaires continues de E .

Munissons E de la topologie faible notée $\sigma(E, E')$ définie comme suit (23 p. 202) :

Pour toute partie finie $(x'_1, \dots, x'_n) \in E'$ et tout $\varepsilon > 0$ on désigne par $V((x'_1, \dots, x'_n), \varepsilon)$ l'ensemble des $x \in E$ tels que $|\langle x, x'_i \rangle| < \varepsilon \quad i = 1, \dots, n$ ($\langle \cdot, \cdot \rangle$ désigne la dualité entre E et E').

Soit B l'ensemble des V . B est un système fondamental de voisinages de 0 pour une topologie localement convexe sur E , compatible avec sa structure d'espace vectoriel. Cette topologie est désignée par $\sigma(E, E')$.

E' est muni de la topologie notée $\sigma(E', E)$ ou topologie faible* (75).

La topologie $\sigma(E, E')$ est associée à la famille de semi normes

$$\{p_{x'}\}_{x' \in E'} : p_{x'}(x) = |\langle x, x' \rangle| \quad \forall x \in E$$

La topologie $\sigma(E', E)$ est associée à la famille de semi normes $\{p_x\}_{x \in E}$:

$$p_x(x') = |\langle x, x' \rangle| \quad \forall x' \in E'.$$

Soit $\{x_n\}$ une suite d'éléments de E . On dit que cette suite converge faiblement vers $x_\infty \in E$ (pour la topologie $\sigma(E, E')$ sur E) si

$$\forall x' \in E' \quad \lim_{n \rightarrow \infty} \langle x_n, x' \rangle = \langle x_\infty, x' \rangle$$

Soit $\{x'_n\}$ une suite d'éléments de E' . On dit que cette suite converge faiblement* (ou faiblement par abus de langage et quand il n'y a aucune ambiguïté) vers $x'_\infty \in E'$ (pour la topologie $\sigma(E', E)$ sur E') si

$$\forall x \in E \quad \lim_{n \rightarrow \infty} \langle x, x'_n \rangle = \langle x, x'_\infty \rangle.$$

Toutes les notations sont classiques. On les trouvera dans les références (3, 21, 22, 34, 40, 47, 52, 75, 76)

I N T R O D U C T I O N

Soit E un espace vectoriel topologique et E' son dual topologique.
Soit $\{x'_n\}$ une suite d'éléments de E' qui converge faiblement vers x'_∞ .

Un problème que l'on rencontre fréquemment en analyse numérique est celui de l'estimation de $\langle x, x'_\infty \rangle$ à partir de certains $\langle x, x'_n \rangle$. Comme nous allons le voir ce problème peut être abordé de deux points de vue qui semblent a priori totalement différents.

Le premier point de vue consiste à remplacer la suite $\{x'_n\}$ par une autre suite faiblement convergente $\{y'_n\}$ telle que $\langle x, y'_n \rangle$ "converge plus vite" que $\langle x, x'_n \rangle$. On effectue cette transformation à l'aide d'une application linéaire T' de E' dans lui-même. Les propriétés de convergence de $\langle x, y'_n \rangle$, d'accélération de la convergence, etc., sont des propriétés de l'opérateur T' et ne dépendent pas des propriétés de E . On a l'habitude d'appeler méthodes de sommation les méthodes définies de cette façon. Il est évident qu'il revient au même de parler de la transformation de x en $y \in E$ à l'aide de l'opérateur T dont T' est le transposé (T est aussi le transposé de T' pour les bonnes topologies). Les propriétés de T' pourront donc s'énoncer sous forme de propriétés de T et inversement. (Il est possible, comme cela est fait au chapitre I, de prendre pour T un opérateur de E dans un autre espace vectoriel topologique). Le second point de vue est un point de vue d'extrapolation. On peut le formuler ainsi :

Soit $\varphi : \mathbb{N} \times D \rightarrow \mathbb{R} \quad D \subseteq \mathbb{R}^k$ telle que

$$\forall a \in D : \lim_{n \rightarrow \infty} \varphi(n, a) = 0.$$

Soit $V_\varphi \{x'_n\}$ la variété de E des x tels que :

$$\langle x, x'_\infty - x'_n \rangle = \varphi(n, a) \quad \forall n \in \mathbb{N}$$

Supposons que $\varphi(n,a)$ ait la propriété suivante :

$\forall \alpha = (\alpha_1, \dots, \alpha_k) \in \mathbb{R}^k$ donné et assez voisin de $0 \in \mathbb{R}^k$
il existe un rang n_0 tel que $\forall n \geq n_0$ le système :

$$\alpha_1 = \varphi(n,a) - \varphi(n+1,a)$$

$$\alpha_2 = \varphi(n+1,a) - \varphi(n+2,a)$$

$$\alpha_k = \varphi(n+k-1,a) - \varphi(n+k,a)$$

admettre une et une seule solution a .

Alors si $x \in \bigcup_p \{x'_n\}$ on aura : a indépendant de n et

$$\langle x, x'_\infty \rangle = \langle x, x'_p \rangle + \varphi(p,a) \quad \forall p \in \mathbb{N} \quad (o-1)$$

Le calcul du vecteur a s'effectue par interpolation. On a :

$$\langle x, x'_n \rangle - \langle x, x'_{n-1} \rangle = \varphi(n-1,a) - \varphi(n,a)$$

Soit n_1 un rang tel que $|\langle x, x'_n \rangle - \langle x, x'_{n-1} \rangle|$ soit suffisamment petit pour tout $n \geq n_1$.

Dès que $n \geq n_0, n_1$ on peut écrire :

$$\alpha_1 = \langle x, x'_{n+1} \rangle - \langle x, x'_n \rangle = \varphi(n,a) - \varphi(n+1,a)$$

$$\alpha_2 = \langle x, x'_{n+2} \rangle - \langle x, x'_{n+1} \rangle = \varphi(n+1,a) - \varphi(n+2,a)$$

$$\alpha_k = \langle x, x'_{n+k} \rangle - \langle x, x'_{n+k-1} \rangle = \varphi(n+k-1,a) - \varphi(n+k,a)$$

On tire de ce système un et un seul vecteur a d'où $\langle x, x'_\infty \rangle$ par extrapolation en appliquant (o-1).

En pratique il est rare que l'on connaisse exactement $\varphi(n,a)$ et par conséquent $x \notin \bigcup_p \{x'_n\}$. La solution a obtenue par interpolation dépendra de n , ce qui donnera :

$$\langle x, x'_\infty \rangle \neq \langle x, x'_p \rangle + \varphi(p, a(n))$$

On peut se demander si $\langle x, x'_p \rangle + \varphi(p, a(n))$ est une meilleure approximation de $\langle x, x'_\infty \rangle$ que $\langle x, x'_p \rangle \quad \forall p \in \mathbb{N}$. x étant donné une autre question est de savoir s'il existe au moins un $y \in E$ tel que :

$$\langle y, x'_p \rangle = \langle x, x'_p \rangle + \varphi(p, a(n)) \quad \forall p \in \mathbb{N}$$

On a reconnu le problème des moments. En supposant qu'il existe au moins un y qui vérifie la relation précédente une seconde question est de savoir si y est plus "voisin" de $\bigvee \varphi \{ x'_n \}$ que ne l'est x , c'est à dire si $\langle y, x'_p \rangle$ converge vers $\langle x, x'_\infty \rangle$ et cela "plus vite" que $\langle x, x'_p \rangle$.

Un exemple classique d'une telle extrapolation est l'extrapolation polynomiale. On a dans ce cas :

$$\varphi(n, a) = \sum_{i=1}^k \frac{a_i}{n^i}$$

Un autre exemple est celui des sommes d'exponentielles :

$$\varphi(n, a) = \sum_{i=1}^k a_i \cdot a_i^n \quad \left| a_i \right| < 1 \quad \forall i$$

Ce point de vue consiste donc à remplacer $x \in E$ par $y \in E$. Il est évidemment souhaitable que $\langle y, x'_n \rangle$ "converge plus vite" que $\langle x, x'_n \rangle$. Les propriétés de convergence de $\langle y, x'_n \rangle$, d'accélération de la convergence, etc., ne sont plus des propriétés de l'opérateur T (non forcément linéaire) défini par $y = T x$ mais des propriétés de son domaine d'application. En effet si x est "voisin" de $\bigvee \varphi \{ x'_n \}$, $a(n)$ ne doit pas dépendre "beaucoup" de n pour $n \geq n_0, n_1$.

Ces méthodes définissent ce qu'il est convenu d'appeler un procédé d'extrapolation. Tels sont les procédés de Romberg et de Richardson basés sur l'extrapolation polynomiale et qui, de ce fait, sont linéaires ou les \mathcal{E} et \mathcal{P} algorithmes de Wynn basés sur l'extrapolation par des sommes d'exponentielles ou par des fractions rationnelles et qui donnent lieu à des procédés non linéaires. Nous montrerons, au paragraphe 1-4 que les procédés de sommation sont des cas particuliers de l'extrapolation par des sommes d'exponentielles. Avant d'étudier ces méthodes nous allons définir un certain nombre de concepts dont nous aurons besoin :

définition 0 - 1 : Soit T un endomorphisme de l'espace vectoriel E ,
 $\{x'_n\}$ une suite d'éléments de E' et x un élément de E .

Si $\lim_{n \rightarrow \infty} \langle Tx, x'_n \rangle$ existe et est finie on dit que x est sommable par rapport à $\{x'_n\}$.

définition 0 - 2 : Soit E un espace vectoriel topologique et E' son dual.
 Soit $\{x'_i\}$ une suite d'éléments de E' qui converge faiblement vers x'_∞ .

On forme si $x \in E$ et $y \neq 0 \in E$:

$$\rho(n; x, y) = \left| \frac{\langle x, x'_n - x'_\infty \rangle}{\langle y, x'_n - x'_\infty \rangle} \right|$$

Posons si elle existe : $\lim_{n \rightarrow \infty} \rho(n; x, y) = e$

- si $0 < e < 1$ on dit que x converge mieux que y par rapport à $\{x'_i\}$
- si $e = 0$ on dit que x converge plus vite que y par rapport à $\{x'_i\}$

définition 0 - 3 : Soient $x' = \{x'_i\}$ et $y' = \{y'_i\}$ deux suites d'éléments de E' qui convergent faiblement vers x'_∞ et y'_∞ .
 On forme si $x \neq 0 \in E$:

$$\rho'(n; x', y') = \left| \frac{\langle x, x'_n - x'_\infty \rangle}{\langle x, y'_n - y'_\infty \rangle} \right|$$

Posons si elle existe : $\lim_{n \rightarrow \infty} \rho'(n; x', y') = e'$

- si $0 < e' < 1$ on dit que $x' = \{x'_i\}$ améliore la convergence de x par rapport à $y' = \{y'_i\}$.
- si $e' = 0$ on dit que x' accélère la convergence de x par rapport à y' .

remarque importante :

Les deux définitions précédentes reviennent au même s'il existe une application linéaire bornée T de E dans lui-même telle que $y = Tx$. On définit alors l'opérateur transposé T' par $\langle Tx, x' \rangle = \langle x, T'x' \rangle$, d'où :
 $\langle y, x'_n - x'_\infty \rangle = \langle Tx, x'_n - x'_\infty \rangle = \langle x, T'(x'_n - x'_\infty) \rangle = \langle x, T'x'_n - T'x'_\infty \rangle = \langle x, y'_n - y'_\infty \rangle$

Dans ce cas il revient au même de définir des relations sur E ou sur E' , cependant celles définies sur E' seront plus générales car elles ne feront pas intervenir d'éléments de E (voir par exemple la définition 0-3).

Si T est un opérateur non linéaire, T' n'existe pas en général et les relations seront à définir sur E . Ce ne seront plus alors des définitions spécifiques de l'opérateur mais des définitions dépendant de $x \in E$ (voir par exemple la définition 0-2).

Définition 0-4 : Si $\lim_{n \rightarrow \infty} \langle x, x'_n \rangle = \lim_{n \rightarrow \infty} \langle Tx, x'_n \rangle \quad \forall x \in E$

on dit que la transformation T est régulière par rapport à $x' = \{x'_n\}$

Définition 0-5 : Si $\langle Tx, x'_n \rangle = \langle x, x'_\infty \rangle \quad \forall n \geq n_0$

on dit que la transformation T est exacte pour $x \quad \forall n \geq n_0$

Définition 0-6 : Si $\langle x, x'_n \rangle = \langle x, x'_\infty \rangle \quad \forall n$ et si $\langle Tx, x'_n \rangle =$

$\langle x, x'_\infty \rangle \quad \forall n$ et x , on dit que T est consistante

Définition 0-7 : Si T est régulière et consistante on dit que T est totale.

Le premier chapitre traite des procédés de sommation, c'est-à-dire des transformations linéaires d'un espace de suites dans lui-même. Je montre que le théorème de Toeplitz qui donne les conditions nécessaires et suffisantes pour que la suite déduite converge vers la même limite que la suite initiale, est une conséquence des théorèmes de Banach - Steinhaus et de prolongement des identités. Après avoir rappelé les principaux procédés de sommation j'étudie en détail la méthode de Baranger pour laquelle j'ai obtenu une valeur optimale du paramètre α . Ensuite je montre que les procédés totaux de sommation sont des extrapolations par des sommes d'exponentielles. Par conséquent les procédés de cette classe peuvent être considérés comme des cas particuliers de l' \mathcal{E} -algorithme et sont donc moins puissants que lui. Le chapitre II est consacré au problème de l'approximation de la fonctionnelle limite sur (C) espaces des suites convergentes. Après avoir formulé ce problème de façon générale j'étudie plusieurs algorithmes particuliers : le procédé de Richardson, les \mathcal{E} et \mathcal{F} -algorithmes

et le procédé d'Aitken étendu qui sont respectivement des extrapolations par des polynômes, des sommes d'exponentielles, des fractions rationnelles et des suites d'ordre quelconque. Après un rappel de certaines propriétés des \mathcal{E} et \mathcal{P} -algorithmes, je caractérise la variété de (C) pour laquelle l' \mathcal{E} -algorithme fournit une valeur exacte de la fonctionnelle limite. Je donne une interprétation de l' \mathcal{E} -algorithme à l'aide des équations aux différences et je démontre sa continuité et sa différentiabilité. Je propose enfin une forme étendue du \mathcal{P} -algorithme ainsi qu'un algorithme plus général : le \mathcal{P}^r -algorithme. Dans le troisième chapitre j'étudie en détail les conditions de convergence et d'accélération de la convergence des \mathcal{E} et \mathcal{P} -algorithmes. Après avoir établi des théorèmes généraux de convergence de ces algorithmes je montre que l' \mathcal{E} -algorithme converge quand la suite initiale est soit totalement monotone, soit totalement oscillante soit totalement oscillante décalée. Je donne ensuite des conditions nécessaires et suffisantes d'accélération de la convergence puis j'introduis un paramètre d'accélération de la convergence ce qui permet d'effectuer la liaison avec le \mathcal{P} -algorithme. Je caractérise la valeur optimale de ce paramètre d'accélération et je montre que ce choix optimal est le seul qui permette d'accélérer la convergence. Après avoir comparé ces algorithmes avec le procédé de Richardson j'explique tous les théorèmes d'accélération ainsi que le paramètre d'accélération à l'aide des développements asymptotiques. Le chapitre IV traite de l'approximation de la fonctionnelle limite sur une certaine classe de fonctions. On utilise pour cela les formes confluentes des \mathcal{E} et \mathcal{P} -algorithmes. Après avoir posé ce problème de façon générale j'étudie les premières et secondes formes confluentes de l' \mathcal{E} -algorithme. Je démontre des théorèmes de convergence et d'accélération de la convergence. Je caractérise la classe des fonctions pour lesquelles ces formes confluentes fournissent la valeur exacte de la fonctionnelle limite. J'introduis dans ces algorithmes un paramètre puis une fonction d'accélération. J'établis chaque fois les théorèmes correspondants. J'effectue enfin la liaison avec le \mathcal{P} -algorithme

et j'explique les résultats obtenus à l'aide des développements asymptotiques. Le chapitre V est consacré à la mise en oeuvre pratique des formes discrètes des algorithmes étudiés. Je formule d'abord les règles particulières à utiliser pour éviter la perte de précision due à la troncature puis je parle de la stabilité et de la propagation des erreurs d'arrondis. On donne divers modes d'emploi des algorithmes.

Le chapitre VI expose les applications et les résultats numériques obtenus. On applique les formes discrètes des algorithmes aux limites des suites et aux sommes de séries. Je démontre que l'application de l' \mathcal{E} -algorithme aux méthodes itératives du premier ordre pour résoudre un système linéaire conduit à la solution exacte et ceci même si la méthode est divergente. Pour les systèmes d'équations non linéaire je donne un nouvel algorithme de résolution. Cette méthode, basée sur l' \mathcal{E} -algorithme, est d'ordre deux et ne nécessite ni calculs de dérivées ni inversions de matrices. On étudie également l'interpolation inverse puis on compare les algorithmes à la méthode de Romberg pour le calcul des intégrales définies et la dérivation numérique. Je propose un nouveau procédé basé sur le \mathcal{E}^2 -algorithme et qui donne de meilleurs résultats que la méthode de Romberg. Les formes confluentes sont testées sur le calcul de la limite d'une fonction, la résolution des équations et le calcul des intégrales impropres. Enfin, je donne une seconde forme discrète de l' \mathcal{E} -algorithme.

C H A P I T R E I

LES PROCÉDES DE SOMMATION

Ce chapitre traite des procédés de sommation, c'est-à-dire des transformations linéaires de suite à suite. On montre que la condition nécessaire et suffisante pour que la suite déduite converge vers la même limite que la suite initiale est une conséquence des théorèmes de Banach - Steinhaus et de prolongement des identités. Après avoir rappelé les principaux procédés de sommation et étudié en détail celui de Baranger, on explique une certaine classe de procédés de sommation comme étant des extrapolations par des sommes d'exponentielles. Par conséquent, les procédés de cette classe peuvent être considérés comme des particuliers de l' ε -algorithme qui sera étudié au chapitre suivant et sont donc moins puissants que lui.

1-1 Généralités

Soient E et F deux espaces de Banach de norme $\| \cdot \|$

E' et F' leurs duals topologiques.

$\{f'_n\}$ une suite de F' faiblement convergente vers $f'_\infty \in F'$.

$\{e'_n\}$ une suite de E' faiblement convergente vers $e'_\infty \in E'$

$\{e\}$ un système total de E c'est-à-dire dont les combinaisons linéaires finies engendrent un sous espace D partout dense dans E.

T un opérateur linéaire continu de E dans F.

T' l'opérateur transposé de T (de F' dans E')

$\{x'_n\}$ la suite d'éléments de E' définie par $x'_n = T' f'_n \quad \forall n$.

Le problème des conditions à imposer à T pour que $\lim_{n \rightarrow \infty} \langle Tx, f'_n \rangle = \lim_{n \rightarrow \infty} \langle x, e'_n \rangle$ peut être remplacé par celui des conditions à imposer à $\{x'_n\}$ pour que ce soit une suite faiblement convergente vers $x'_\infty \in E'$ et pour que $\lim_{n \rightarrow \infty} \langle x, x'_n \rangle = \lim_{n \rightarrow \infty} \langle x, e'_n \rangle$. Nous adopterons ce second point de vue. Rappelons d'abord quelques

résultats :

Théorème I-1 (35) : Une condition nécessaire et suffisante pour que $\{x'_n\}$ converge faiblement vers $x'_\infty \in E'$ et que :

$$1^\circ) - \|x'_n\| < M \quad \forall n$$

$$2^\circ) - \langle e'_k, x'_n - x'_\infty \rangle \text{ converge vers } 0 \quad \forall k$$

Théorème I-2 (21) : Soit $\{y'_n\}$ une suite d'éléments de E' qui converge faiblement vers y'_∞ . Si $\lim_{k \rightarrow \infty} \langle x, y'_k \rangle = \lim_{k \rightarrow \infty} \langle x, e'_k \rangle \quad \forall x \in D$ alors $\lim_{k \rightarrow \infty} \langle x, y'_k \rangle =$

$$\lim_{k \rightarrow \infty} \langle x, e'_k \rangle \quad \forall x \in E.$$

Démonstration :

$$\forall x \in \bar{D} = E \exists \{x_n\} \in D : \lim_{n \rightarrow \infty} x_n = x. \text{ or } \langle x_n, y'_\infty \rangle = \langle x_n, e'_\infty \rangle \quad \forall n$$

donc $\langle x, y'_\infty \rangle = \langle x, e'_\infty \rangle \quad \forall x \in E$ puisque y'_∞ et e'_∞ sont continues.

(voir dans Dieudonné (21) le principe de prolongement des identités)

Soit c l'espace des suites convergentes de nombres réels. Muni de la norme

$$\|x\| = \sup_i |x_i| \quad (\text{où } x_i \text{ désigne la } i\text{ème composante de la suite } x) \text{ c est}$$

un espace de Banach.

Le dual de c est l'espace des suites de nombres réels telles que $\sum_{i=1}^{\infty} |x'_i|$ converge. On peut le munir de la norme $\|x'\| = \sum_{i=1}^{\infty} |x'_i|$. C'est alors un espace de Banach.

Le sous espace D engendré par le système libre $e_0 = (1, 1, \dots)$, $e_1 = (1, 0, 0 \dots)$, $e_2 = (0, 1, 0 \dots)$, etc. est dense dans c . Prenons $E = F = c$ et $e'_n = f'_n =$ forme linéaire qui à une suite associe sa nième composante.

L'opérateur T est défini par une matrice infinie $A = (a_{ij})$ et T' est défini par une matrice A' transposée de A : $A' = (a_{ji})$.

Les conditions du théorème I-1 s'écrivent :

$$1^\circ) - \|x'_n\| = \|T' f'_n\| = \sum_k |a_{nk}| < M \quad \forall n$$

$$2^\circ) - \lim_{n \rightarrow \infty} \langle e_k, x'_n \rangle = \lim_{n \rightarrow \infty} \langle T e_k, f'_n \rangle = \lim_{n \rightarrow \infty} a_{nk} = b_k \quad \forall k > 0$$

$$\lim_{n \rightarrow \infty} \langle e_0, x'_n \rangle = \lim_{n \rightarrow \infty} \langle T e_0, f'_n \rangle = \lim_{n \rightarrow \infty} \sum_k a_{nk} = b_0$$

d'où le théorème suivant :

Théorème I-3 : Une condition nécessaire et suffisante pour que la matrice A définisse un endomorphisme de c est que :

$$1^\circ) \sum_k |a_{nk}| < M \quad \forall n$$

$$2^\circ) \lim_{n \rightarrow \infty} a_{nk} = b_k \quad \forall k$$

$$3^\circ) \lim_{n \rightarrow \infty} \sum_k a_{nk} = b_0$$

Si $b_k = 0 \quad \forall k$ et $b_0 = 1$ alors les conditions du théorème I-2 sont vérifiées car $\lim_{n \rightarrow \infty} \langle e_k, e'_n \rangle = 0 \quad \forall k > 0$ et $\lim_{n \rightarrow \infty} \langle e_0, e'_n \rangle = 1$. D'où le résultat :

définition I-1 : Soit C_s la variété des suites qui convergent vers S .

On dit que la matrice A définit une méthode régulière de sommation si elle définit un endomorphisme de C_s quelquesoit S .

Théorème I-4 : (théorème de Toeplitz) : une condition nécessaire et suffisante pour que la matrice A définisse une méthode régulière de sommation est que :

$$1^\circ) \quad \sum_k |a_{nk}| < M \quad \forall n$$

$$2^\circ) \quad \lim_{n \rightarrow \infty} a_{nk} = 0 \quad \forall k$$

$$3^\circ) \quad \lim_{n \rightarrow \infty} \sum_k a_{nk} = 1$$

1-2 Principales méthodes

Il est évident que les méthodes régulières sont les plus intéressantes car elles préservent la limite. Enumérons maintenant les principales :

Moyennes de Hölder :

Les éléments de la matrice A sont :

$$\left. \begin{array}{l} a_{ij} = \frac{1}{i} \quad j = 1, \dots, i \\ a_{ij} = 0 \quad j > i \end{array} \right\} i = 1, \dots, \infty$$

Cette matrice A définit la méthode appelée $(H,1)$. La méthode (H,k) k entier > 1 est définie par la matrice A^k .

Moyennes de Cesaro :

La matrice A définissant le procédé noté (c,k) est donnée par $A = H L^{k-1}$ où H est identique à la matrice A de la méthode $(H,1)$ et où L est une matrice infinie triangulaire inférieure dont tous les termes valent 1. Il est évident que les méthodes $(H,1)$ et $(C,1)$ sont identiques. On montre que les procédés (c,k) et (H,k) sont réguliers et que si $x \in S$ (espace des suites de nombres réels) est sommable par (c,k) il l'est par (H,k) et inversement.

Méthodes d'Euler :

Elles sont définies par une matrice A dépendant d'un paramètre positif q :

$$\left. \begin{aligned} a_{ij} &= \frac{1}{(q+1)^{i+1}} \binom{i+1}{j+1} q^{i-j} & j = 1, \dots, i \\ a_{ij} &= 0 & j > i \end{aligned} \right\} i = 1, \dots, \infty$$

Cette matrice définit une méthode notée (E,q). La plus connue des méthodes d'Euler est celle avec q = 1. Ces méthodes sont régulières et telles que si $x \in S$ est sommable par (E,q') elle est sommable vers la même limite par (E, q) $\forall q > q'$.

Il existe de nombreuses autres méthodes de sommation. Elles ont été étudiées en détail par Hardy (33).

1-3 Accélération de la convergence

Il est en général difficile de caractériser les suites qui peuvent être accélérées par un procédé de sommation régulier déterminé et ceci même si la matrice correspondante est triangulaire inférieure comme c'est le cas pour les méthodes de Hölder, Cesaro et Euler. Cependant j'ai pu effectuer cette étude dans un cas simple :

Baranger (4) a étudié les p_i et les A_i ($i = 0, \dots, n$) qui rendent la formule :

$$\sum_{i=0}^{\infty} u_i = \sum_{i=0}^n A_i u_{p_i}$$

optimale en un certain sens pour un sous-ensemble donné de l'ensemble des séries absolument convergentes. Soit H_a l'espace des suites :

$$H_a = \left\{ u = \{u_n\} \mid \sum_0^{\infty} |u_n| < \infty \text{ et } \sum_0^{\infty} \frac{(\Delta u_n)^2}{a^n} < \infty \right\}$$

avec $0 < a < 1$ et $\Delta u_n = u_{n+1} - u_n$. C'est un espace de Hilbert inclus topologiquement dans l^1 .

Baranger a montré que pour $p_i = i$ le meilleur choix des A_i était donné par :

$$A_0 = A_1 = \dots \dots \dots A_{n-1} = 1 \quad \text{et} \quad A_n = \frac{1}{1-a}$$

Nous nous proposons d'étudier plus en détail cette transformation et, plus particulièrement, de caractériser les séries pour lesquelles $V_n = \sum_{i=0}^n A_i u_i$

converge vers $\sum_{i=0}^{\infty} u_i$ plus vite que $S_n = \sum_{i=0}^n u_i$. Il nous faut d'abord montrer que la transformation $\{S_n\} \rightarrow \{V_n\}$ est régulière au sens de Toeplitz. On vérifie immédiatement que :

$$V_n = A_n S_n + (1-A_n) S_{n-1} \quad n = 1, \dots \quad (I-1)$$

Par conséquent, la matrice de Toeplitz correspondant à cette transformation de suite à suite est telle que :

$$a_{ii} = A_n \quad \text{et} \quad a_{i+1,i} = 1-A_n \quad \text{pour} \quad i = 1, \dots$$

tous les autres termes étant nuls. Les trois conditions de Toeplitz du théorème I-4 :

$$\sum_k |a_{nk}| < M \quad \forall n$$

$$\lim_{n \rightarrow \infty} a_{nk} = 0 \quad \forall k$$

$$\lim_{n \rightarrow \infty} \sum_k a_{nk} = 1$$

sont bien vérifiées et on a donc $\lim_{n \rightarrow \infty} V_n = \sum_{i=0}^{\infty} u_i$

supposons maintenant que $0 < a_1 < a_2 < 1$. On a $H_{a_1} \subset H_{a_2}$. En effet

$$\frac{1}{a_1^n} > \frac{1}{a_2^n} \quad \text{et}$$

$$\sum_0^{\infty} \frac{(\Delta u_n)^2}{a_2^n} \quad \circ \quad \sum_0^{\infty} \frac{(\Delta u_n)^2}{a_1^n}$$

donc si $u = \{u_n\} \in H_{a_1}$ alors $u \in H_{a_2}$ car la série est absolument convergente

D'autre part si $u \in H_{a_1}$ et si $\forall \varepsilon > 0 \quad u \notin H_{a_1 - \varepsilon}$ alors $u \in H_a \quad \forall a \in [a_1, 1[$.

peut être existe t-il une valeur de a optimale dans $[a_1, 1[$, c'est-à-dire telle que $\{V_n\}$ converge plus vite que $\{S_n\}$ au sens de la définition 0-2 ? C'est

cette valeur que nous allons maintenant caractériser.

$$\lim_{n \rightarrow \infty} \frac{V_{n+1} - V_n}{S_{n+1} - S_n} = 0$$

d'où

$$\lim_{n \rightarrow \infty} \frac{A_{n+1} S_{n+1} + (1-A_{n+1}) S_n - A_n S_n - (1-A_n) S_{n-1}}{S_{n+1} - S_n} =$$

$$\lim_{n \rightarrow \infty} \frac{A (S_{n+1} - 2 S_n + S_{n-1}) + S_n - S_{n-1}}{S_{n+1} - S_n} \quad \text{puisque } A_n = A_{n+1} = \frac{1}{1-a} = A. \text{ D'où}$$

$$\lim_{n \rightarrow \infty} \frac{A (\Delta S_n - \Delta S_{n-1}) + \Delta S_{n-1}}{\Delta S_n} = \lim_{n \rightarrow \infty} \left[A + (1-A) \frac{\Delta S_{n-1}}{\Delta S_n} \right] = 0$$

ce qui donne :

$$\lim_{n \rightarrow \infty} \frac{\Delta S_{n-1}}{\Delta S_n} = \frac{A}{A-1} = \frac{1}{a}$$

C'est la condition nécessaire et suffisante pour que $\{V_n\}$ converge plus vite que $\{S_n\}$, ce qui nous donne comme valeur optimale pour a :

$$a_{\text{opt}} = \lim_{n \rightarrow \infty} \frac{\Delta S_n}{\Delta S_{n-1}} = \lim_{n \rightarrow \infty} \frac{u_{n+1}}{u_n}$$

Par conséquent plus $|a - a_{\text{opt}}|$ sera faible et plus la convergence sera améliorée.

Si $a = a_{\text{opt}}$ alors la convergence sera accélérée. On voit également que si a_{opt}

est négatif, il est impossible d'accélérer la convergence puisque a doit

positif. Il en est de même si a_{opt} vaut 1.

Prenons maintenant un exemple. Soit la série de terme général $u_n = br^n$ avec $|r| < 1$
 on a $S = \sum_0^{\infty} u_n = \frac{b}{1-r}$

$$\sum_0^{\infty} \frac{(\Delta u_n)^2}{a^n} = \sum_0^{\infty} \frac{b^2 r^{2n}}{a^n} = \sum_0^{\infty} b^2 \left(\frac{r^2}{a}\right)^n = \frac{b^2}{1 - \frac{r^2}{a}} \quad \text{si } r^2 < a < 1$$

et par conséquent $u = \{u_n\} \in H_a$ pour $a \in]r^2, 1[$. On a $\frac{u_{n+1}}{u_n} = r$ d'où $a_{opt} = r$.

On voit que si $0 < r < 1$ $a_{opt} \in]r^2, 1[$ et peut donc être atteint. Par contre si $-1 < r < 0$, a_{opt} est négatif et ne peut être atteint; la valeur de a qui améliorera le plus la convergence est, dans ce cas, $a = 0$ qui est la plus voisine de a_{opt} et correspond à la suite initiale $\{S_n\}$.

Pour $b = 1$ et $r = 0,5$ on a $S = 2$ et les résultats suivants :

suite initiale	$a = 0,25$	$a = 0,5$	$a = 0,6$
1			
1,5	1,6666667	2,0000000	2,2500000
1,75	1,8333333	2,0000000	2,1250000
1,875	1,9166667	2,0000000	2,0625000
1,9375	1,9583333	2,0000000	2,0312500

Pour $b = 1$ et $r = -0,5$ on a $S = 0,66666667$ et les résultats suivants :

Suite initiale	$a = 0,25$	$a = 0,5$	$a = 0,6$
1			
0,5	0,33333334	0	- 0,24999997
0,75	0,83333333	1,0000000	1,1250000
0,625	0,58333334	0,50000000	0,43750000
0,6875	0,70833333	0,75000000	0,78125000

Au lieu d'appliquer la méthode de Baranger au calcul de la somme d'une série, on peut l'appliquer au calcul de la limite d'une suite en utilisant (I-1) et la relation $a_{opt} = \lim_{n \rightarrow \infty} \frac{\Delta S_n}{\Delta S_{n-1}}$. Prenons, par exemple, une suite de la forme

$S_n = S + \alpha \lambda^n$. On trouve $a_{opt} = \lambda$.

Prenons $S = 1, \alpha = 2$ et $\lambda = 0,9$. On trouve $A_{opt} = 10$. Nous avons effectué les calculs avec $A = 9, 10$ et 11 et obtenu les résultats suivants :

Suite initiale	A = 9	A = 10	A = 11
2,8			
2,62	1,1799998	0,99999976	0,81999993
2,458	1,1619999	1,000000	0,83800006
2,3122	1,1457999	1,000000	0,85419989

En fait si l'on considère (I-1) comme un procédé de sommation au sens défini précédemment, il n'y a aucune obligation d'avoir $0 < a < 1$, la seule restriction à imposer est $a \neq 1$. Ainsi le a_{opt} peut toujours être atteint sauf s'il vaut 1.

Pour les autres procédés de sommation les résultats connus sont beaucoup plus partiels. C'est ainsi que pour la méthode d'Euler avec $q = 1$ on sait (5) que la transformation de la série $x = x_0 - x_1 + x_2 - x_3 + \dots$ en la série $y_0 + y_1 + y_2 + y_3 + \dots$ avec $y_n = \Delta^n x_0 / 2^{n+1}$ accélère la convergence si :

- la série initiale est alternée $x_i \geq 0$
- la suite x_i est totalement monotone c'est-à-dire si $(-1)^k \Delta^k x_n \geq 0$ pour $k = 0, 1, \dots$ et quelque soit n
- $x_{i+1} / x_i > a \geq 1/2$

Sous ces conditions on a

$$\left| \frac{Y_n - x}{X_n - x} \right| \leq 2^{-n} a^{-n-1} \quad \text{avec } X_n \text{ et } Y_n$$

étant les sommes partielles des séries; on voit que pour les séries telles que

$$\lim_{n \rightarrow \infty} \frac{x_{n+1}}{x_n} = 1 \text{ on a } a = 1 \text{ et que } \sum_n y_n \text{ converge au moins aussi rapidement que } \sum_n 2^{-n}.$$

Dans d'autres cas la série de terme général y_n converge moins vite que la série initiale. Ainsi $x_i = r^i$ implique $y_i = 2^{-i-1} (1-r)^i$ et $|r| < |1-r|/2$ pour

$$-1 < r < 1/3.$$

On voit combien ces résultats sont partiels et l'intérêt qu'il y aurait à trouver pour tous les procédés de sommation des résultats analogues à celui que nous avons obtenu pour la méthode de Baranger.

1-4 Interprétation des procédés de sommation

D'après la définition 0-7, nous dirons qu'un procédé de sommation est total s'il est régulier et consistant. Dans la définition 0-6 prenons pour x'_n la forme linéaire qui à une suite fait correspondre son $n^{\text{ième}}$ terme. Nous avons vu qu'un procédé de sommation était défini par la donnée d'une matrice $A = (a_{ij})$. D'où le résultat suivant :

Théorème 1-5 :

Une condition nécessaire et suffisante pour que la matrice $A = (a_{ij})$ définisse une méthode consistante de sommation est que :

$$\sum_{j=0}^{\infty} a_{ij} = 1 \quad \forall i$$

Nous appellerons cette condition : condition de consistance. La démonstration découle immédiatement de la définition 0-6. Ce théorème exprime simplement le fait que si la suite initiale est consistante, la suite obtenue par application du procédé de sommation défini par la matrice A est identique à la suite initiale.

L'étude du procédé total particulièrement simple de Baranger donné par (I-1) m'a permis de trouver une interprétation de tous les procédés totaux de sommation tels que je viens de les définir. Ces procédés sont des extrapolations par des sommes d'exponentielles complexes où les exposants sont des quantités fixées spécifiques de la méthode et où les seules inconnues sont les coefficients de la combinaison linéaire de ces exponentielles.

Considérons, en effet, le procédé (I-1). Au lieu de prendre $a = a_{opt} = \lim_{n \rightarrow \infty} \frac{\Delta S_n}{\Delta S_{n-1}}$ on peut songer à prendre comme approximation de a_{opt} la quantité

$\frac{\Delta S_n}{\Delta S_{n-1}}$. On obtient alors :

$$V_n = \frac{S_n}{1 - \frac{\Delta S_n}{\Delta S_{n-1}}} + \left(1 - \frac{1}{1 - \frac{\Delta S_n}{\Delta S_{n-1}}}\right) S_{n-1}$$

$$= \frac{S_{n-1} \cdot S_{n+1} - S_n^2}{\Delta^2 S_{n-1}} = \mathcal{E}_2^{(n-1)}$$

ce qui n'est autre que le procédé Δ^2 d'Aitken habituel (voir paragraphe 2.3.1). Ainsi la méthode de Baranger apparaît comme un cas particulier du Δ^2 obtenu en prenant pour a une valeur constante. En effet, comme nous l'avons vu, le procédé (I-1) donne S si l'on a $S_n = S + \alpha a^n$ où a est le paramètre de la méthode, tandis que le procédé Δ^2 d'Aitken donne S si $S_n = S + \alpha \lambda^n$ où λ est quelconque et indépendant de la méthode. L'ensemble des suites pour lesquelles (I-1) donne la valeur de S est donc un sous-ensemble de celles pour lesquelles le procédé Δ^2 donne S . (I-1) sera donc un accélérateur de convergence moins puissant que le procédé Δ^2 . Généralisons (13) :

Soit un procédé total de sommation défini par une matrice $A = (a_{ij})$. Nous supposons que cette matrice est telle que $\forall i \exists k(i) : \forall j > k(i) \ a_{ij} = 0$.

Soit $\{x_n\}$ une suite infinie convergente $\lim_{n \rightarrow \infty} x_n = S$ et soit $\{y_n\}$ la suite obtenue en appliquant à $\{x_n\}$ le procédé de sommation, on a :

$$y_i = \sum_{j=1}^{k(i)} a_{ij} x_{j+n}$$

Pour i fixé y_i sera égal à S $\forall n$ si et seulement si l'équation aux différences

$$S = \sum_{j=1}^{k(i)} a_{ij} x_{j+n} \quad (I-2)$$

est vérifiée $\forall n$. La solution générale de cette équation aux différences est une combinaison linéaire d'exponentielles identiques à celle obtenue pour l' \mathcal{E} -algorithme. (voir chapitre II).

on a :

$$x_n = S + \varphi(n, a)$$

où $\varphi(n, a)$ est la fonction correspondant à l' ε -algorithme avec $k = k(i) - 1$

(voir théorème II-2) et où les s_i sont liés aux racines de l'équation caractéristique de l'équation aux différences homogène de (I-2) :

$$a_{i1} + a_{i2} t + \dots + a_{i, k(i)} t^{k(i)-1} = 0$$

d'où le théorème suivant :

Théorème I-6 :

Soit un procédé total de sommation défini par la matrice infinie $A = (a_{ij})$ où

$\forall i \exists k(i) : \forall j > k(i)$ on ait $a_{ij} = 0$. Soit $\{x_n\}$ une suite convergente telle que $\lim_{n \rightarrow \infty} x_n = S$.

Une condition nécessaire et suffisante pour que $\exists i : \forall n \sum_{j=1}^{k(i)} a_{ij} x_{j+n} = S$

est que :

$$1^\circ) x_n = S + \sum_{i=1}^p A_i(n) r_i^n + \sum_{i=p+1}^q [B_i(n) \cos b_i n + c_i(n) \sin b_i n] e^{s_i n} + \sum_{i=1}^m c_i \delta_{in} \quad \forall n \geq 0$$

où $A_i, B_i,$ et C_i sont des polynômes en n tels que si d_i est égal au degré de A_i plus un pour $i = 1, \dots, p$ et au plus grand des degrés de B_i et de C_i plus un pour $i = p+1, \dots, q$ on ait :

$$m + \sum_{i=1}^p d_i + 2 \sum_{i=p+1}^q d_i = k(i) - 1$$

2°) le polynôme

$$\sum_{j=1}^{k(i)} a_{ij} t^{j-1}$$

admette les p racines réelles distinctes r_i pour $i=1, \dots, p$

les $2(q-p)$ racines complexes distinctes $e^{s_i} (\cos b_i \pm i \sin b_i)$

pour $i = p+1, \dots, q$

m racines nulles

Les conséquences de ce théorème sont importantes. On remarque d'abord que les r_i , les s_i et les b_i sont fixés et spécifiques du procédé de sommation employé. Les seules inconnues sont les coefficients des polynômes A_i , B_i , et C_i , les c_i et S . Ce qui explique pourquoi les procédés de sommation sont linéaires, tandis que l' \mathcal{E} -algorithme qui détermine également les s_i et les b_i est non linéaire. On voit également que si x_n est de la forme donnée dans l'énoncé du théorème I-6 il faudra $k(i)$ termes pour déterminer les inconnues par un procédé linéaire de sommation et $2k(i) - 1$ termes pour les déterminer par l' \mathcal{E} -algorithme. Mais la conclusion la plus importante est la suivante :

Tout procédé total de sommation est un cas particulier d'un procédé où les r_i , les s_i et les b_i seraient également des inconnues. Un tel procédé existe : c'est l' \mathcal{E} -algorithme. Par conséquent tout procédé total de sommation donnera de moins bons résultats que l' \mathcal{E} -algorithme.

Enfin ceci nous montre que le procédé de Richardson et celui de Romberg étant des procédés totaux de sommation, sont des extrapolations. Il en est de même de la méthode de Baranger, ce qui explique pourquoi nous l'avons retrouvée comme cas particulier du procédé Δ^2 d'Aitken, ainsi que des procédés de Cesaro et de Hölder.

Notons encore que si le procédé n'est pas total, S n'est pas solution particulière de l'équation inhomogène et qu'alors il n'existe pas de suites tendant vers S et telles que le procédé soit exact pour ces suites.

CHAPITRE II

LES PROCÉDES D'EXTRAPOLATION

Ce chapitre est consacré au problème de l'extrapolation des suites. Après avoir formulé celui-ci de façon générale on étudie le procédé de Richardson, les ε et ρ -algorithmes et le procédé d'Aitken étendu qui sont respectivement des extrapolations par des polynômes, des sommes d'exponentielles, des fractions rationnelles et des suites d'ordre quelconque. L' ε -algorithme est relié à la table de Padé, aux fractions continues et à la classe des algorithmes de losange pour lesquels on donne les équations aux dérivées partielles qui leur sont associées. On établit une interprétation de l' ε -algorithme à l'aide des équations aux différences et l'on démontre sa continuité et sa différentiabilité en un sens que l'on précise. On propose enfin une forme étendue du ρ -algorithme ainsi qu'un algorithme plus général : le ρ^r algorithme.

2-1 Le problème de l'extrapolation

Reprenons le problème de l'extrapolation tel qu'il a été posé dans l'introduction mais en le généralisant quelque peu. Il s'agit d'étudier la nature des éléments $y \in E$ pour lesquels on sait calculer $\langle y, y'_\infty \rangle$ à partir de certains $\langle y, y'_n \rangle$. Soit $\varphi(x, a)$ une fonction donnée :

$$\varphi: \mathbb{R} \times D \rightarrow \mathbb{R} \quad D \subseteq \mathbb{R}^k \quad \text{telle que} \quad \forall a \in D \quad \text{on ait}$$

$$\lim_{x \rightarrow \infty} \varphi(x, a) = 0.$$

Le problème est de savoir si, $\forall \alpha = (\alpha_1, \dots, \alpha_k) \in \mathbb{R}^k$ et assez voisin de $0 \in \mathbb{R}^k$, il existe un n_0 tel que $\forall x_{k+1} > x_k > \dots > x_1 > x_{n_0}$ le système :

$$\alpha_1 = \varphi(x_1, a) - \varphi(x_2, a)$$

$$\alpha_2 = \varphi(x_2, a) - \varphi(x_3, a)$$

$$\alpha_k = \varphi(x_k, a) - \varphi(x_{k+1}, a)$$

admette une et une seule solution a .

Posons $\Psi_i(x, a) = \varphi(x_i, a) - \varphi(x_{i+1}, a) \quad i = 1, \dots, k$

$$\Psi(x, a) = (\Psi_1(x, a), \dots, \Psi_k(x, a)) \in \mathbb{R}^k$$

$$x = (x_1, \dots, x_{k+1}) \in \mathbb{Q} \times \mathbb{R}^{k+1}$$

Ψ est une application de $\mathbb{Q} \times D \rightarrow \mathbb{R}^k$ et le système précédent devient :

$$\Psi(x, a) = \alpha$$

Des conditions suffisantes d'existence et d'unicité d'une solution a de cette équation nous sont données par le théorème d'existence des fonctions implicites (voir par exemple L. SCHWARTZ (50) pages 278 et 293).

Nous sommes ici dans le cas où les espaces sont de dimensions finies et où l'espace des a et celui des α ont même dimension. On peut, par conséquent énoncer le

théorème II-1

Si les fonctions $\Psi_i(X, a)$ sont des fonctions scalaires continues des variables x_k et a_i et si le déterminant jacobien des $\frac{\partial \Psi_i}{\partial a_j}(Y, b)$ est différent de zéro dans un certain voisinage du point (Y, b) tel que $\Psi(Y, b) = \alpha$, alors on pourra, dans ce voisinage, déterminer a comme fonction implicite de X .

Il est donc du plus grand intérêt de rechercher les fonctions $\Psi(x, a)$ qui vérifient ce théorème. Soit, en effet, $\{y'_n\}$ une suite d'éléments de E' qui converge faiblement vers y'_{∞} . Soit $V_\varphi\{y'_n\}$ la variété de E des y tels que : $\langle y, y'_{\infty} - y'_n \rangle = \Psi(x_n, a) \quad \forall n \in \mathbb{N}$ où les x_n forment une suite strictement monotone. On a $\forall y \in V_\varphi\{y'_n\}$:

$$\langle y, y'_n \rangle - \langle y, y'_{n-1} \rangle = \Psi(x_{n-1}, a) - \Psi(x_n, a)$$

Si n_1 est un rang tel que $|\langle y, y'_n \rangle - \langle y, y'_{n-1} \rangle|$ soit assez petit pour tout $n \geq n_1$, alors, dès que $n \geq n_0, n_1$, on aura :

$$\langle y, y'_{n+1} \rangle - \langle y, y'_n \rangle = \Psi(x_n, a) - \Psi(x_{n+1}, a) \tag{I}$$

$$\langle y, y'_{n+k} \rangle - \langle y, y'_{n+k-1} \rangle = \Psi(x_{n+k-1}, a) - \Psi(x_{n+k}, a)$$

On tire de ce système un et un seul vecteur a si les conditions du théorème II-1 sont vérifiées. D'où

$$\langle y, y'_{\infty} \rangle = \langle y, y'_p \rangle + \Psi(x_p, a) \quad \forall p \in \mathbb{N}$$

Malheureusement il est souvent difficile de savoir si la condition du théorème II-1 est remplie et, même si c'est le cas, il n'est pas sûr que la solution du système (I) puisse se mettre sous forme explicite. D'autre part si $y \notin V_\varphi\{y'_n\}$ la solution a de (I) dépendra de x_n, \dots, x_{n+k} . Il est donc important de

savoir si cette dépendance est telle que si y est "voisin" de $V_{\varphi}\{y'_n\}$, la solution a ne dépend pas "beaucoup" de x_n, \dots, x_{n+k} .

on aura alors :

$$\langle y, y'_{\infty} \rangle \neq \langle y, y'_p \rangle + \varphi(x_p, a) = \langle z, y'_p \rangle \quad \forall p \in \mathbb{N}$$

Un second problème est de savoir si $\exists z$ qui vérifie l'égalité précédente et si z converge plus vite que y au sens de la définition 2.

Nous allons maintenant étudier quatre procédés d'extrapolation basés sur des fonctions $\varphi(x, a)$ particulières. Pour les trois premiers la condition du théorème II-1 est vérifiée et l'on connaît explicitement la solution du système I. Ce sont : la méthode de Richardson basée sur l'interpolation polynomiale, l' ε -algorithme basé sur l'interpolation par des sommes d'exponentielles et le p -algorithme basé sur l'interpolation par des fractions rationnelles. Pour le quatrième procédé il n'a pas été possible de vérifier la condition du théorème II-1 ni de trouver explicitement la solution du système I. Ce procédé est construit à partir de $\varphi(x, a) = a_1 a_2^{a_3 x}$ avec $0 < a_2 < 1$ et $a_3 \geq 1$.

2-2 Le procédé d'extrapolation de Richardson

L'extrapolation par un polynôme a été étudiée par Richardson (46) puis reprise en détail par Laurent (36).

soit $\varphi(x, a)$ la fonction :

$$\varphi(x, a) = \sum_{j=1}^k \frac{a_j}{x^j}$$

et soit $\{x_n\}$ une suite positive strictement croissante tendant vers l'infini avec n . On a :

$$\varphi_i(x, a) = \sum_{j=1}^k a_j \left(\frac{1}{x_i^j} - \frac{1}{x_{i+1}^j} \right)$$

$$\frac{\partial \varphi_i}{\partial a_j} = \frac{1}{x_i^j} - \frac{1}{x_{i+1}^j}$$

On sait que le déterminant jacobien est différent de zéro puisque les x_n sont tous distincts et, par conséquent, les conditions du théorème II-1 sont toujours vérifiées.

La résolution du système (I) conduit à :

$$\langle Y, Y'_{\infty} \rangle = \sum_{j=1}^{k+1} A_j^{k+1} \langle Y, Y'_{n+j-1} \rangle$$

avec

$$A_j = \prod_{\substack{i=1 \\ i \neq j}}^{k+1} \frac{1}{\frac{x_i}{x_j}}$$

si $Y \notin V_{\varphi}\{Y'_n\}$ Laurent a démontré qu'une condition nécessaire et suffisante pour que ce procédé soit régulier est qu'il existe $\alpha > 1$ tel que l'on ait,

$$\forall k, \quad \frac{x_{k+1}}{x_k} \geq \alpha.$$

Plus généralement, soit $g(x)$ une fonction numérique strictement décroissante, soit $\varphi(x, a) = \sum_{j=1}^k a_j (g(x))^j$ et soit $\{x_n\}$ une suite strictement croissante.

Une condition est nécessaire et suffisante pour que le procédé d'extrapolation défini par $\sum_{i=1}^{k+1} B_i^{k+1} \langle Y, Y'_{n+i-1} \rangle$ soit régulier $\forall Y \notin V_{\varphi}\{Y'_n\}$ est qu'il existe $\alpha > 1$ tel que l'on ait, $\forall k$:

$$\frac{g(\infty) - g(x_k)}{g(\infty) - g(x_{k+1})} \geq \alpha.$$

Si l'on prend $g(x) = \frac{1}{x^2}$ et $x_{k+1} = 2x_k$ on retrouve le procédé d'extrapolation de Romberg qui est particulièrement bien adapté à l'accélération des suites obtenues par intégration approchée d'une fonction $y \in C^{\infty}[0, 1]$ par la méthode des trapèzes.

2-3 L' ε - algorithme

Je vais maintenant étudier en détail l' ε -algorithme. Cet algorithme a été trouvé par Wynn (64). C'est une méthode qui permet de mettre en oeuvre la généralisation du procédé bien connu d'accélération de la convergence qu' est le Δ^2 d'Aitken. Ce procédé a été imaginé par A.C. Aitken (2) en 1926. Il fut repris en 1939 par Schmidt (49), en 1952 par Lubkin (38) puis généralisé en 1955 par Shanks (51). En 1956, Wynn a proposé une méthode récursive pour effectuer cette transformation de Shanks : c'est l' ε -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 (6), en 1957, a mis en évidence, à l'aide des fractions continues de Stieltjes, les relations qui existent entre l' ε -algorithme, l'algorithme QD de Rutishauser (48) et une classe plus générale d'algorithmes: les algorithmes de losange. Depuis divers auteurs comme Clark (16,17), Gray, Atchison, Adams et Schucany (26, 27, 28, 29, 30, 31) ont étudié la convergence et l'application au calcul d'intégrales impropres de transformations dérivées du procédé classique d'Aitken. Ils n'ont malheureusement pas pu généraliser leurs méthodes et cela à cause de la difficulté de démonstrations. En effet, contrairement au procédé de Richardson, tous les algorithmes que nous allons maintenant étudier, donnent $\langle y, y'_\infty \rangle$ comme une fonction non linéaire de certains $\langle y, y'_n \rangle$.

2-3-1 Le procédé Δ^2 d'Aitken et l' ε -algorithme

soit $\varphi: \mathbb{N} \times D \rightarrow \mathbb{R}$ (ou \mathbb{C}) $D \subset \mathbb{R}^2$ (ou \mathbb{C}^2)

$$\varphi(n, a) = a_1 \cdot a_2^n \quad \text{avec } a_1 \neq 0 \text{ et } |a_2| < 1$$

le système :

$$\alpha_1 = \varphi(n, a) - \varphi(n+1, a) = a_1 a_2^n (1-a_2)$$

$$\alpha_2 = \varphi(n+1, a) - \varphi(n+2, a) = a_1 a_2^{n+1} (1-a_2)$$

admet une et une seule solution car les conditions du théorème II-1 sont toujours vérifiées $\forall a \in D$. De plus :

$$\begin{vmatrix} \varphi(n,a) & \varphi(n+1,a) \\ \varphi(n+1,a) & \varphi(n+2,a) \end{vmatrix} = 0 \quad (\text{II-1})$$

Soit $\{x'_n\}$ une suite d'éléments de E' qui converge faiblement vers x'_∞ et soit $x \in V_\varphi \{x'_n\}$. Alors on a :

$$\langle x, x'_\infty \rangle = \langle x, x'_n \rangle - \frac{\langle x, \Delta x'_n \rangle^2}{\langle x, \Delta^2 x'_n \rangle} \quad (\text{II-2})$$

avec $\langle x, \Delta x'_n \rangle = \langle x, x'_{n+1} - x'_n \rangle = \langle x, x'_{n+1} \rangle - \langle x, x'_n \rangle$

et $\langle x, \Delta^{k+1} x'_n \rangle = \langle x, \Delta^k x'_{n+1} - \Delta^k x'_n \rangle$.

En effet une combinaison linéaire des lignes et des colonnes de (II-1) nous donne :

$$\begin{vmatrix} \varphi(n,a) & \varphi(n+1,a) - \varphi(n,a) \\ \varphi(n+1,a) - \varphi(n,a) & \varphi(n+2,a) - 2\varphi(n+1,a) + \varphi(n,a) \end{vmatrix} = 0$$

or $\varphi(n,a) - \varphi(n+1,a) = \langle x, \Delta x'_n \rangle$ d'où :

$$\begin{vmatrix} \langle x, x'_\infty - x'_n \rangle & - \langle x, \Delta x'_n \rangle \\ - \langle x, \Delta x'_n \rangle & \langle x, \Delta^2 x'_n \rangle \end{vmatrix} = 0$$

ce qui donne (II-2). C'est le procédé Δ^2 d'Aitken.

soit maintenant $\varphi : \mathbb{N} \times D \rightarrow \mathbb{R}$ (ou \mathbb{C}) $D \subset \mathbb{R}^{2k}$ (ou \mathbb{C}^{2k})

$$\varphi(n,a) = \sum_{i=1}^k a_i a_{k+i}^n$$

avec $a_i \neq 0$ et $|a_{k+i}| < 1$ $i = 1, \dots, k$.

on a : $\lim_{n \rightarrow \infty} \varphi(n, a) = 0$

et :

$$\left| \begin{array}{l} \varphi(n, a) \quad \varphi(n+1, a) \quad \text{-----} \quad \varphi(n+k, a) \\ \varphi(n+1, a) \quad \varphi(n+2, a) \quad \text{-----} \quad \varphi(n+k+1, a) \\ \text{-----} \\ \varphi(n+k, a) \quad \varphi(n+k+1, a) \quad \text{-----} \quad \varphi(n+2k, a) \end{array} \right| = 0 \quad (\text{II-3})$$

d'où si $x \in V_{\varphi} \{x'_n\}$ une et une seule solution :

$$\left| \begin{array}{l} \langle x, x'_n \rangle \quad \langle x, \Delta x'_n \rangle \quad \text{-----} \quad \langle x, \Delta^k x'_n \rangle \\ \langle x, \Delta x'_n \rangle \quad \langle x, \Delta^2 x'_n \rangle \quad \text{-----} \quad \langle x, \Delta^{k+1} x'_n \rangle \\ \text{-----} \\ \langle x, \Delta^k x'_n \rangle \quad \langle x, \Delta^{k+1} x'_n \rangle \quad \text{-----} \quad \langle x, \Delta^{2k} x'_n \rangle \end{array} \right| \quad (\text{II-4})$$

$$\langle x, x'_\infty \rangle = \left| \begin{array}{l} \langle x, \Delta^2 x'_n \rangle \quad \text{-----} \quad \langle x, \Delta^{k+1} x'_n \rangle \\ \text{-----} \\ \langle x, \Delta^{k+1} x'_n \rangle \quad \text{-----} \quad \langle x, \Delta^{2k} x'_n \rangle \end{array} \right|$$

Cette solution est obtenue en effectuant des différences de lignes et de colonnes dans (II-3), puis en remplaçant $\varphi(n, a)$ par $\langle x, x'_\infty \rangle - \langle x, x'_n \rangle \forall n$ (64).

Si $x \notin V_{\varphi} \{x'_n\}$, pour k fixé la relation (II-4) n'est plus égale à $\langle x, x'_\infty \rangle$ mais à une certaine quantité $e_k(\langle x, x'_n \rangle)$. Nous ne nous posons pas pour l'instant la question de savoir s'il existe au moins un $y \in E$ tel que $\langle y, x'_n \rangle = e_k(\langle x, x'_n \rangle)$ et si cet y est plus "voisin" de $V_{\varphi} \{x'_n\}$ que ne l'est x .

Ne considérons la transformation $e_k(\langle x, x'_n \rangle)$ de Shanks que comme une application de E dans l'espace des suites d'éléments de \mathbb{R} (ou \mathbb{C}) et examinons la question du

calcul de $e_k (<x, x'_n >)$. On voit que l'application de la relation (II-4) nécessite le calcul de déterminants d'ordre élevé. Le but de l' \mathcal{E} -algorithme de Wynn est de fournir un procédé récursif qui permet d'éviter ce travail.

Posons : $\mathcal{E}_{-1}^{(n)} = 0 \quad n = 1, 2, \dots$

$$\mathcal{E}_0^{(n)} = <x, x'_n > \quad n = 0, 1, \dots$$

et employons la forme progressive de l' \mathcal{E} -algorithme :

$$\mathcal{E}_{k+1}^{(n)} = \mathcal{E}_{k-1}^{(n+1)} + \frac{1}{\mathcal{E}_k^{(n+1)} - \mathcal{E}_k^{(n)}} \quad k, n = 0, 1, \dots \quad (II-5)$$

Si les quantités $\mathcal{E}_k^{(n+1)}$ et $\mathcal{E}_k^{(n)}$ deviennent égales ou très voisines on emploie alors des règles particulières déduites de (II-5) afin d'éviter une perte de précision due à la troncature; nous les étudierons au chapitre V. On voit que l' \mathcal{E} -algorithme (II-5) permet de construire, à partir d'un nombre fini de valeurs initiales $\mathcal{E}_0^{(n)}$, un tableau triangulaire à double entrée analogue à une table de différences. L'indice inférieur représente le numéro de la colonne et l'indice supérieur celui de la diagonale :

	$\mathcal{E}_0^{(0)}$					
$\mathcal{E}_{-1}^{(1)}$	$\mathcal{E}_0^{(1)}$	$\mathcal{E}_1^{(0)}$				
$\mathcal{E}_{-1}^{(2)}$	$\mathcal{E}_0^{(2)}$	$\mathcal{E}_1^{(1)}$	$\mathcal{E}_2^{(0)}$			
$\mathcal{E}_{-1}^{(3)}$	$\mathcal{E}_0^{(3)}$	$\mathcal{E}_1^{(2)}$	$\mathcal{E}_2^{(1)}$	$\mathcal{E}_3^{(0)}$		
$\mathcal{E}_{-1}^{(4)}$	$\mathcal{E}_0^{(4)}$	$\mathcal{E}_1^{(3)}$	$\mathcal{E}_2^{(2)}$	$\mathcal{E}_3^{(1)}$	$\mathcal{E}_4^{(0)}$	
$\mathcal{E}_{-1}^{(5)}$	$\mathcal{E}_0^{(5)}$	$\mathcal{E}_1^{(4)}$	$\mathcal{E}_2^{(3)}$	$\mathcal{E}_3^{(2)}$	$\mathcal{E}_4^{(1)}$	$\mathcal{E}_5^{(0)}$
	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots

Wynn a montré (64) que $\mathcal{E}_{2k}^{(n)} = e_k (<x, x'_n >)$ et que $\mathcal{E}_{2k+1}^{(n)} = 1/e_k (<x, \Delta x'_n >)$.

Par conséquent, seules les colonnes d'indice pair sont intéressantes, les colonnes impaires étant des calculs intermédiaires. On voit que la relation (II-5) relie dans le tableau \mathcal{E} des valeurs situées aux quatre angles d'un

losange. L' \mathcal{E} -algorithme fait partie d'une classe d'algorithmes appelés algorithmes de losange. Des relations existent entre certains de ces algorithmes et notamment entre l' \mathcal{E} -algorithme et l'algorithme QD.

2-3-2 Propriétés de l' \mathcal{E} -algorithme

L' \mathcal{E} -algorithme est un algorithme non linéaire. C'est-à-dire que si, partant de quantités $\mathcal{E}_0^{(n)}$ on obtient $\mathcal{E}_k^{(n)}$ et si, partant de $\bar{\mathcal{E}}_0^{(n)}$ on obtient $\bar{\mathcal{E}}_k^{(n)}$, alors en partant de $\mathcal{E}_0^{(n)} + \bar{\mathcal{E}}_0^{(n)}$ on n'obtiendra pas $\mathcal{E}_k^{(n)} + \bar{\mathcal{E}}_k^{(n)}$;

L' \mathcal{E} -algorithme est cependant linéaire par rapport à la multiplication par un scalaire et par rapport à l'addition d'un scalaire, du moins en ce qui concerne les colonnes paires qui seules nous intéressent. Si, partant de $\mathcal{E}_0^{(n)}$ on obtient $\mathcal{E}_k^{(n)}$ alors, partant de $\bar{\mathcal{E}}_0^{(n)} = a \mathcal{E}_0^{(n)} + b$, on obtiendra

$$\bar{\mathcal{E}}_{2k}^{(n)} = a \mathcal{E}_{2k}^{(n)} + b \text{ et } \bar{\mathcal{E}}_{2k+1}^{(n)} = \mathcal{E}_{2k+1}^{(n)} / a.$$

On montre également (59), que si on applique l' \mathcal{E} -algorithme à des quantités $\mathcal{E}_0^{(n)} = \langle x, x'_n \rangle$ qui vérifient :

$$\sum_{i=0}^k c_i \langle x, x'_{n+i} \rangle = b \quad n = 0, 1, \dots$$

et si les racines $\lambda_1, \dots, \lambda_k$ de l'équation $\sum_{i=0}^k c_i \lambda^i = 0$ sont réelles, distinctes et telle que :

$$|\lambda_1| > |\lambda_2| > \dots > |\lambda_j| > 1 > |\lambda_{j+1}| > \dots > |\lambda_k|.$$

alors :

$$\lim_{n \rightarrow \infty} \frac{\mathcal{E}_{2i-1}^{(n+1)}}{\mathcal{E}_{2i-1}^{(n)}} = \frac{1}{\lambda_i}$$

et

$$\lim_{n \rightarrow \infty} \frac{\mathcal{E}_{2i-2}^{(n+1)}}{\mathcal{E}_{2i-2}^{(n)}} = \begin{cases} \lambda_i & i = 1, 2, \dots, j \\ 1 & i = j+1, \dots, k \end{cases}$$

Si, de plus $\sum_{i=0}^k c_i \neq 0$ alors :

$$\varepsilon_{2k+1}^{(n)} = \infty \quad \text{et} \quad \varepsilon_{2k}^{(n)} = \frac{b}{\sum_{i=0}^k c_i} \quad n = 0, 1, \dots$$

Cette dernière propriété est liée à la fonction $\varphi(n, a)$ correspondant à l' ε -algorithme. Nous avons pu démontrer (15) que la fonction $\varphi(n, a)$ de l' ε -algorithme est d'une classe plus générale que la classe des fonctions de la forme $\sum_{i=1}^k a_i a_{k+i}^n$ qui avait été considérée jusque là. Etablissons le résultat suivant :

Théorème II-2

La fonction $\varphi(n, a)$ correspondant à l' ε -algorithme est de la forme :

$$\begin{aligned} \varphi(n, a) = & \sum_{i=1}^p A_i(n) r_i^n + \sum_{i=p+1}^q (B_i(n) \cos b_i n + C_i(n) \sin b_i n) e^{s_i n} \\ & + \sum_{i=0}^m c_i \delta_{in} \end{aligned}$$

A_i, B_i et C_i sont des polynômes en n tels que si d_i est égal au degré de A_i plus un pour $i=1, \dots, p$ et au plus grand des degrés de B_i et de C_i plus un pour $i = p+1, \dots, q$ on ait :

$$m + 1 + \sum_{i=1}^p d_i + 2 \sum_{i=p+1}^q d_i = k$$

avec la convention que $m = -1$ s'il n'y a aucun terme en δ_{in} (symbole de Kronecker). Le vecteur a est formé par toutes les inconnues : r_i, s_i, b_i, c_i ainsi que les coefficients des polynômes A_i, B_i et C_i .

démonstration : une condition nécessaire et suffisante pour que le déterminant (II-3) soit nul est qu'il existe $\alpha_0, \dots, \alpha_k$ tels que :

$$\alpha_0 \varphi(n+k, a) + \dots + \alpha_k \varphi(n, a) = 0$$

$$\alpha_0 \varphi(n+2k, a) + \dots + \alpha_k \varphi(n+k, a) = 0$$

donc $\varphi(n, a), \varphi(n+1, a), \dots, \varphi(n+k, a)$ sont solution de l'équation aux

différences.

$$\alpha_0 y_{n+k} + \dots + \alpha_k y_n = 0$$

Les solutions de cette équation aux différences forment un espace vectoriel de dimension inférieure ou égale à k . Par conséquent $\varphi(n,a), \dots, \varphi(n+k,a)$ sont linéairement dépendants et il existe des constantes c_1, \dots, c_k telles que :

$$\varphi(n+k,a) + c_1 \varphi(n+k-1,a) + \dots + c_k \varphi(n,a) = 0$$

On sait (voir par exemple (32)) que la solution générale de cette équation aux différences est de la forme donnée dans l'énoncé du théorème. On remarque que le déterminant (II-3) est nul même si les s_i sont positifs et, par conséquent, la fonction $\varphi(n,a)$ divergente. Cela signifie que l'application de l' \mathcal{E} -algorithme à une suite initiale de la forme $\mathcal{E}_0^{(n)} = s + \varphi(n,a)$ nous donnera $\mathcal{E}_{2k}^{(n)} = s \quad \forall n = 0, 1, \dots$ et cela même si $\lim_{n \rightarrow \infty} \varphi(n,a) = \pm\infty$. Le pseudo-polynôme en $\delta : \sum_{i=0}^m c_i \delta_{in}$, montre, d'autre part, que les $m+1$ premières quantités initiales $\mathcal{E}_0^{(0)}, \mathcal{E}_0^{(1)}, \dots, \mathcal{E}_0^{(m)}$ peuvent être absolument quelconques puisqu'aucune condition n'est imposée sur les scalaires c_0, \dots, c_m . Au chapitre VI nous verrons que les méthodes linéaires itératives du premier ordre pour résoudre les systèmes linéaires générant des suites appartenant à $V_\varphi \{x'_n\}$, φ étant la fonction du théorème II-2, et donc que l' \mathcal{E} -algorithme donne la solution exacte du système. Si φ est une application de $\mathbb{R} \times D \rightarrow \mathbb{R}$, alors pour des points équidistants x_n donnés par $x_n = x_0 + nh$ on se ramène à la forme précédente par le changement de variable $x = (x - x_0) / h$. L' \mathcal{E} -algorithme (II-5) est par conséquent encore valable dans ce cas.

Signalons enfin que les quantités $\mathcal{E}_k^{(n)}$ peuvent s'exprimer en fonction des déterminants de Hankel. Soit s_n, s_{n+1}, \dots une suite de scalaires. On définit :

$$H_k^{(n)}(s_n) = \frac{\begin{vmatrix} s_n & s_{n+1} & \dots & s_{n+k-1} \\ s_{n+1} & s_{n+2} & \dots & s_{n+k} \\ \dots & \dots & \dots & \dots \\ s_{n+k-1} & s_{n+k} & \dots & s_{n+2k-2} \end{vmatrix}}{\begin{vmatrix} s_n & \Delta s_n & \dots & \Delta^{k-1} s_n \\ \Delta s_n & \Delta^2 s_n & \dots & \Delta^k s_n \\ \dots & \dots & \dots & \dots \\ \Delta^{k-1} s_n & \Delta^k s_n & \dots & \Delta^{2k-2} s_n \end{vmatrix}}$$

Les déterminants de Hankel $H_k^{(n)}(S_n)$ sont liés par la relation de récurrence :

$$H_k^{(n+1)}(S_{n+1}) \cdot H_k^{(n-1)}(S_{n-1}) - \left[H_k^{(n)}(S_n) \right]^2 = H_{k+1}^{(n-1)}(S_{n-1}) \cdot H_{k-1}^{(n+1)}(S_{n+1})$$

avec $H_0^{(n)}(S_n) = 1$ et $H_1^{(n)}(S_n) = S_n$

Wynn (64) a montré que :

$$\varepsilon_{2k}^{(n)} = \frac{H_{k+1}^{(n)}(\langle x, x'_n \rangle)}{H_k^{(n)}(\langle x, \Delta^2 x'_n \rangle)}$$

$$\varepsilon_{2k+1}^{(n)} = \frac{H_k^{(n)}(\langle x, \Delta^3 x'_n \rangle)}{H_{k+1}^{(n)}(\langle x, \Delta x'_n \rangle)}$$

on a également (67) :

$$\varepsilon_{2k+2}^{(n)} = \varepsilon_{2k}^{(n)} - \frac{\left[H_{k+1}^{(n)}(\langle x, \Delta x'_n \rangle) \right]^2}{H_{k+1}^{(n)}(\langle x, \Delta^2 x'_n \rangle) \cdot H_k^{(n)}(\langle x, \Delta^2 x'_n \rangle)}$$

Ce dernier résultat s'obtient facilement en utilisant un développement de Schweins (1) de la différence :

$$\frac{H_{k+2}^{(n)}(S_n)}{H_{k+1}^{(n)}(\Delta^2 S_n)} - \frac{H_{k+1}^{(n)}(S_n)}{H_k^{(n)}(\Delta^2 S_n)}$$

Cette démonstration nous permet d'ailleurs de trouver entre les déterminants de Hankel la nouvelle relation suivante :

$$H_k^{(n-1)}(\Delta^2 S_{n-1}) \cdot H_k^{(n-1)}(S_{n-1}) - \left[H_k^{(n-1)}(\Delta S_{n-1}) \right]^2 = H_{k+1}^{(n-1)}(S_{n-1}) \cdot H_{k-1}^{(n-1)}(\Delta^2 S_{n-1})$$

on a donc $\varepsilon_{2k}^{(n)} = \frac{H_{k+1}^{(n)}(S_n)}{H_k^{(n)}(\Delta^2 S_n)}$ en posant $\langle x, x'_n \rangle = S_n$

l' ε -algorithme ne peut donc s'appliquer qu'aux suites telles que $H_k^{(n)}(\Delta^2 S_n) \neq 0$

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

Soit H l'espace vectoriel des suites infinies et convergentes de nombres réels ou complexes, telles que $H_k^{(n)}(\Delta^2 S_n) \neq 0 \quad \forall k, n=0,1,\dots$

Soit G_k le sous espace de H des suites finies de $2k+1$ termes et soit E_k l'application qui à $x = (S_n, \dots, S_{n+2k}) \in G_k$ fait correspondre $\varepsilon_{2k}^{(n)} = H_{k+1}^{(n)}(S_n) / H_k^{(n)}(\Delta^2 S_n) \in \mathbb{R}$ (ou \mathbb{C}).

$H_{k+1}^{(n)}(S_n)$ est une application continue par rapport à S_n, \dots, S_{n+2k} , il en est de même de $H_k^{(n)}(\Delta^2 S_n)$. Par conséquent, puisque $H_k^{(n)}(\Delta^2 S_n) \neq 0$, l'application E_k est continue dans G_k . D'où le résultat fondamental suivant :

Théorème II-3 :

L' ε -algorithme est continu.

$H_{k+1}^{(n)}(S_n)$ et $H_k^{(n)}(\Delta^2 S_n)$ sont des applications différentiables et par conséquent E_k est différentiable dans G_k d'où :

Théorème II-4 :

l' ε -algorithme est différentiable

$$\text{on a : } \frac{\partial \varepsilon_{2k}^{(n)}}{\partial S_j} = \frac{H_k^{(n)}(\Delta^2 S_n) \cdot \frac{\partial}{\partial S_j} H_{k+1}^{(n)}(S_n) - H_{k+1}^{(n)}(S_n) \cdot \frac{\partial}{\partial S_j} H_k^{(n)}(\Delta^2 S_n)}{[H_k^{(n)}(\Delta^2 S_n)]^2}$$

pour $j = n, \dots, n+2k$.

Nous allons maintenant donner une interprétation de l' ε -algorithme différente de celle exposée dans l'introduction de ce chapitre.

Soit F_k l'application de H dans S espaces des suites, définie par :

$$F_k : \{u_n\} \in H \longrightarrow \left\{ \sum_{i=0}^k \alpha_i u_{n+i-c} \right\} \in S$$

$$\text{on a : Ker } (F_k) = \left\{ \{u_n\} \mid u_n = \frac{c}{\sum_{i=0}^k \alpha_i} + \varphi(n,a) \quad \forall n \right\}$$

où $\varphi(n,a)$ est la fonction correspondant à l' ε -algorithme (théorème II.2).

Etant donné une suite $(S_n, \dots, S_{n+2k}) \in G_k$, l' ε -algorithme revient à chercher les coefficients $\alpha_0, \dots, \alpha_k$ de l'équation aux différences vérifiée par la suite unique $\{u_n\} \in \text{Ker}(F_k)$ telle que $u_i = S_i$ pour $i = n, \dots, n+2k$ puis à calculer la quantité $S = \frac{c}{\sum_{i=0}^k \alpha_i} = \varepsilon_{2k}^{(n)}$

Nous allons montrer que cet algorithme est équivalent à l' ε -algorithme, que la quantité S est indépendante de c et que la condition $H_k^{(n)}(\Delta^2 S_n) \neq 0$ est identique à $\sum_{i=0}^k \alpha_i \neq 0$.

Pour que $u_i = S_i$ pour $i = n, \dots, n+2k$ avec $\{u_n\} \in \text{Ker}(F_k)$ il faut que :

$$\alpha_0 S_n + \alpha_1 S_{n+1} + \dots + \alpha_k S_{n+k} = c$$

$$\alpha_0 S_{n+1} + \alpha_1 S_{n+2} + \dots + \alpha_k S_{n+k+1} = c$$

$$\alpha_0 S_{n+k} + \alpha_1 S_{n+k+1} + \dots + \alpha_k S_{n+2k} = c$$

où c est une constante non nulle. Dans ce système remplaçons la seconde ligne par sa différence avec la première, la troisième par sa différence avec la seconde, etc..., il vient :

$$\alpha_0 S_n + \dots + \alpha_k S_{n+k} = c$$

$$\alpha_0 \Delta S_n + \dots + \alpha_k \Delta S_{n+k} = 0$$

$$\alpha_0 \Delta S_{n+k-1} + \dots + \alpha_k \Delta S_{n+2k-1} = 0$$

faisons maintenant la même chose avec les colonnes, on obtient :

$$\beta_0 S_n + \beta_1 \Delta S_n + \dots + \beta_k \Delta S_{n+k-1} = c$$

$$\beta_0 \Delta S_n + \beta_1 \Delta^2 S_n + \dots + \beta_k \Delta^2 S_{n+k-1} = 0$$

$$\beta_0 \Delta S_{n+k-1} + \beta_1 \Delta^2 S_{n+k-1} + \dots + \beta_k \Delta^2 S_{n+2k-2} = 0$$

$$\text{avec } \beta_j = \sum_{i=j}^k \alpha_i \quad \text{pour } j = 0, \dots, k$$

$$\text{d'où } \beta_0 = \sum_{i=0}^k \alpha_i = c \cdot \frac{H_k^{(n)}(\Delta^2 S_n)}{H_{k+1}^{(n)}(S_n)} \quad \text{et } S = \frac{c}{\sum_{i=0}^k \alpha_i} = \frac{H_{k+1}^{(n)}(S_n)}{H_k^{(n)}(\Delta^2 S_n)} = \mathcal{E}_{2k}^{(n)}$$

Si le déterminant du système est nul on prend $c \stackrel{i=0}{=} 0$ d'où $S = 0$.

On voit que $H_k^{(n)}(\Delta^2 S_n) \neq 0$ entraîne $\sum_{i=0}^k \alpha_i \neq 0$ et réciproquement. De plus,

si l'on remplace c par $a c$ où a est une constante non nulle, alors les α_i sont remplacés par $a \alpha_i$, et, par conséquent, S demeure inchangé. On peut également démontrer l'identité des deux algorithmes de la façon suivante :

$$\alpha_0 S_n + \dots + \alpha_k S_{n+k} = S \sum_{i=0}^k \alpha_i$$

$$\alpha_0 S_{n+k} + \dots + \alpha_k S_{n+2k} = S \sum_{i=0}^k \alpha_i$$

ou encore :

$$\alpha_0 (S_n - S) + \dots + \alpha_k (S_{n+k} - S) = 0$$

$$\alpha_0 (S_{n+k} - S) + \dots + \alpha_k (S_{n+2k} - S) = 0$$

d'où, pour avoir une solution différente de la solution nulle :

$$\begin{vmatrix} S_n - S & \dots & S_{n+k} - S \\ \dots & \dots & \dots \\ S_{n+k} - S & \dots & S_{n+2k} - S \end{vmatrix} = 0$$

ce qui donne :

$$S = \frac{H_{k=1}^{(n)}(S_n)}{H_k^{(n)}(\Delta^2 S_n)} = \mathcal{E}_{2k}^{(n)}$$

2-3-3 Connection avec la table de Padé

Soit la série $\sum_{i=0}^{\infty} c_i x^i$. Si tous les déterminants de Hankel $H_k^{(n)}(c_n)$

sont différents de zéro, alors il est possible de construire une suite double $P_{ij}(x)$ $i, j = 0, 1, \dots$ de fonctions rationnelles de x . $P_{ij}(x)$ est le quotient de deux polynômes en x ; un numérateur de degré j et un dénominateur de degré i . Ce quotient $P_{ij}(x)$ est caractérisé par le fait que son développement en puissances croissantes de x est identique à la série jusqu'au $(i+j)$ ième terme

compris

$$P_{ij}(x) = \frac{\sum_{k=0}^j a_k x^k}{\sum_{k=0}^i b_k x^k} = \sum_{k=0}^{i+j} c_k x^k + \sum_{k=i+j+1}^{\infty} d_k x^k$$

avec, en général, $c_k \neq d_k$ pour $k > i+j$

On place ces quotients $P_{ij}(x)$ dans une table à double entrée où le premier indice désigne la ligne et le second la colonne. C'est la table de Padé de la série. On montre que l'application de l' \mathcal{E} -algorithme avec $\mathcal{E}_0^{(n)} = \sum_{k=0}^n c_k x^k$ $n = 0, 1, \dots$ donne :

$$\mathcal{E}_{2k}^{(n)} = P_{k, n+k}(x)$$

L' \mathcal{E} -algorithme permet donc de construire ainsi la moitié supérieure de la table de Padé de la série. On peut en construire la seconde moitié de la façon suivante.

On a :

$$\mathcal{E}_{2k+1}^{(n-1)} - \mathcal{E}_{2k-1}^{(n)} = \left[\mathcal{E}_{2k}^{(n)} - \mathcal{E}_{2k}^{(n-1)} \right]^{-1}$$

$$\mathcal{E}_{2k+1}^{(n)} - \mathcal{E}_{2k-1}^{(n+1)} = \left[\mathcal{E}_{2k}^{(n+1)} - \mathcal{E}_{2k}^{(n)} \right]^{-1}$$

d'où en soustrayant et en remplaçant les quantités d'indice inférieur impair

par leur valeur :

$$\left[\begin{matrix} \varepsilon_{2k+2}^{(n-1)} & - \varepsilon_{2k}^{(n)} \end{matrix} \right]^{-1} - \left[\begin{matrix} \varepsilon_{2k}^{(n)} & - \varepsilon_{2k-2}^{(n+1)} \end{matrix} \right]^{-1} = \left[\begin{matrix} \varepsilon_{2k}^{(n+1)} & - \varepsilon_{2k}^{(n)} \end{matrix} \right]^{-1} - \left[\begin{matrix} \varepsilon_{2k}^{(n)} & - \varepsilon_{2k}^{(n-1)} \end{matrix} \right]^{-1}$$

Posons $k = i$ et $m+k=j$ $i, j=0, 1, \dots$ on obtient :

$$\left[P_{i+1, j} - P_{ij} \right]^{-1} - \left[P_{ij} - P_{i-1, j} \right]^{-1} = \left[P_{i, j+1} - P_{ij} \right]^{-1} - \left[P_{ij} - P_{i, j-1} \right]^{-1} \quad (II-6)$$

Ce qui permet de construire toute la table de Padé en partant des conditions aux limites intérieures :

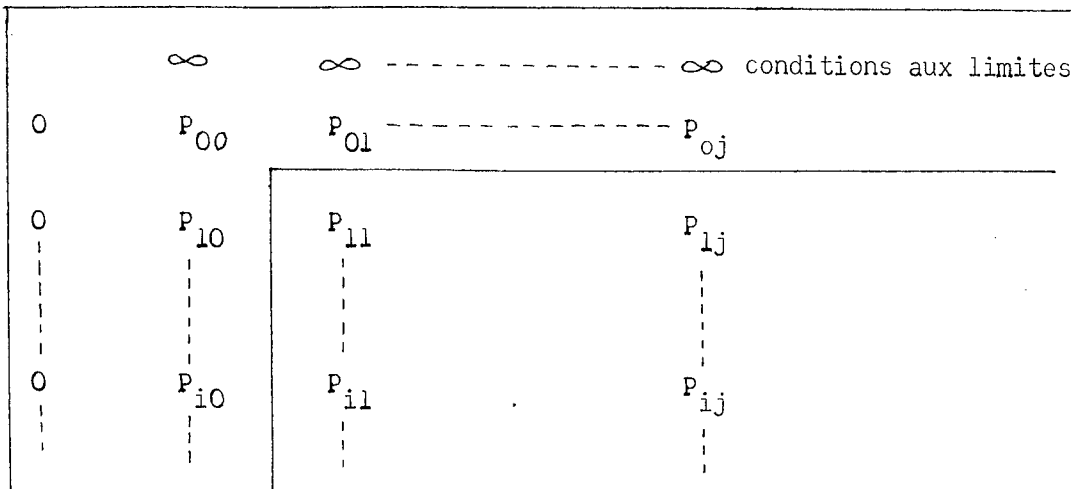
$$P_{0j} = \sum_{k=0}^j c_k x^k \quad j = 0, 1, \dots$$

$$P_{i0} = \left(\sum_{k=0}^i q_k x^k \right)^{-1} \quad i = 0, 1, \dots$$

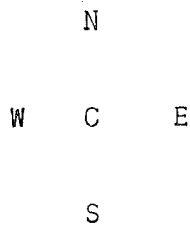
avec

$$\left(\sum_{k=0}^{\infty} c_k x^k \right) \cdot \left(\sum_{k=0}^{\infty} q_k x^k \right) = 1$$

et des conditions aux limites extérieures : $P_{-1, j} = \infty$ $P_{i, -1} = 0$ $i, j=0, 1, \dots$
d'où la table de Padé :



La relation (II-6) relie les quatre sommets d'un losange et son centre



Elle s'écrit $(C - N)^{-1} + (C - S)^{-1} = (C - W)^{-1} + (C - E)^{-1}$

Si quatre des fonctions sont données on peut calculer la cinquième. Etudions le cas où la fonction inconnue n'est pas c. On pose :

$$h(d, e, f) = \frac{\begin{vmatrix} c & c & e \\ f & c & c \\ f & d & e \end{vmatrix}}{\begin{vmatrix} 1 & 1 & 1 \\ c & e & d \\ f & c & d \end{vmatrix}}$$

on a :

$$\begin{aligned}
 N &= h(S, E, W) = h(S, W, E) \\
 E &= h(W, N, S) = h(W, S, N) \\
 S &= h(N, E, W) = h(N, W, E) \\
 W &= h(E, N, S) = h(E, S, N)
 \end{aligned}$$

En un sens N et S sont conjugués ainsi que W et E. Soit, en effet, f une fonction reliant E, C et W. On a $N = f(S)$ et $S = f(N)$. De même, si g est une fonction reliant N, C et S on a $E = g(W)$ et $W = g(E)$. De plus si a est une fonction reliant C et S et b une fonction reliant C et W, on a $N = a(E, W) = a(W, E)$, $E = b(N, S) = b(S, N)$ etc.

La formule (II.6) peut conduire à une indétermination si la fonction inconnue est c. Si on se donne N, S, W, et E alors, en général, c n'est pas unique. Soit z une valeur possible de c. On a :

$$(N+S-E-W) z^2 + 2(NS - EW) z + (NES + NWS - NEW - ESW) = 0$$

Soit Δ le discriminant de cette équation :

$$\Delta = 4 (E - N) (E - S) (W - N) (W - S)$$

C n'est unique que si Δ est nul donc les valeurs de deux sommets voisins égales.

On peut maintenant résoudre le problème suivant ; étant donnés quatre nombres réels $N, E, S,$ et W quelle condition doit-on avoir pour que ces nombres soient les quotients de Padé $N, E, S,$ et W dérivés d'une série entière réelle ?

Puisque dans ce cas C doit être également réel, il faut que Δ soit positif. Une condition suffisante pour cela est que l'une des relations suivantes soit vérifiée ;

$$\begin{aligned} E, W &> \eta_0 = \max (N, S) \\ E, W &< \eta_1 = \min (N, S) \\ \eta_1 &< E, \quad W < \eta_0 \\ E &> \eta_0 \quad \text{et} \quad W < \eta_1 \\ W &> \eta_0 \quad \text{et} \quad E < \eta_1 \end{aligned}$$

Si N, E, S et W sont des fonctions rationnelles alors Δ est le carré d'une fonction rationnelle. Le cas singulier $\varepsilon_{p-1}^{(n+1)} = \varepsilon_{p-1}^{(n)}$ correspond pour $p = 2k$ à C infini. On a dans ce cas la règle singulière $N + S - E - W = 0$.

Bien que ces dernières considérations semblent fort éloignées de l' ε -algorithme, ce sont elles qui ont permis à Wynn de déduire les règles particulières à appliquer dans le cas où $\varepsilon_k^{(n)}$ et $\varepsilon_k^{(n+1)}$ sont égaux ou très voisins (60, 70).

2-3-4 Connection avec les fractions continues

La connection entre l' ε -algorithme et les fractions continues a été mise en évidence par Wynn (74). Une fraction continue est une expression de la forme :

$$b_0 + \frac{a_1}{b_1 + \frac{a_2}{b_2 + \frac{a_3}{b_3 + \dots}}}$$

Pour des raisons typographiques évidentes on introduit les deux notations :

$$b_0 + \frac{a_1}{b_1 + \frac{a_2}{b_2 + \frac{a_3}{b_3} + \dots}} \text{ ou encore}$$

$$b_0 + \frac{a_1}{b_1} + \frac{a_2}{b_2} + \frac{a_3}{b_3} + \dots$$

C'est cette dernière notation que nous utiliserons. Les nombres a_k et b_k s'appellent éléments, a_k/b_k est le $k^{\text{ième}}$ quotient partiel, a_k est le $k^{\text{ième}}$ numérateur partiel et b_k est le $k^{\text{ième}}$ dénominateur partiel. La quantité $c_n = b_0 + \frac{a_1}{b_1} + \dots + \frac{a_n}{b_n}$ est le $n^{\text{ième}}$ approximant (ou convergent) de la fraction continue. Si $\lim_{n \rightarrow \infty} c_n$ existe et est finie, on dit que la fraction continue converge et que sa valeur est $C = \lim_{n \rightarrow \infty} C_n$. On peut calculer C_n de deux manières différentes :

$$D_0 = b_n$$

$$D_{k+1} = b_{n-k-1} + \frac{a_{n-k}}{D_k} \quad k = 0, \dots, n-1$$

on aura :

$$D_n = C_n$$

Une façon plus économique est de calculer les convergents successifs :

$$A_{-1} = 1 \qquad A_0 = b_0$$

$$B_{-1} = 0 \qquad B_0 = 1$$

$$A_n = b_n A_{n-1} + a_n A_{n-2}$$

$$B_n = b_n B_{n-1} + a_n B_{n-2}$$

$$C_n = A_n / B_n \qquad n = 0, 1, \dots$$

Il faut enfin savoir qu'il est possible, en multipliant les numérateurs et les dénominateurs des quotients partiels par des réels non nuls, de transformer la fraction continue en une fraction continue équivalente ayant la même suite des

convergent. Etant donné une certaine fraction continue posons :

$$A'_n = A_{2n} \quad \text{et} \quad B'_n = B_{2n}$$

on a :

$$A_{2n} = b_{2n} A_{2n-1} + a_{2n} A_{2n-2}$$

$$A_{2n-1} = b_{2n-1} A_{2n-2} + a_{2n-1} A_{2n-3}$$

$$A_{2n-2} = b_{2n-2} A_{2n-3} + a_{2n-2} A_{2n-4}$$

Multiplions la première de ces égalités par b_{2n-2} , la seconde par $b_{2n} \cdot b_{2n-2}$,

la dernière par $-a_{2n-1} \cdot b_{2n}$ et faisons la somme. On obtient :

$$b_{2n-2} A_{2n} = (a_{2n} b_{2n-2} + b_{2n} b_{2n-1} b_{2n-2} + a_{2n-1} b_{2n}) A_{2n-2} - a_{2n-1} a_{2n-2} b_{2n} A_{2n-4}$$

d'où :

$$A'_n = \frac{a_{2n} b_{2n-2} + b_{2n} b_{2n-1} b_{2n-2} + a_{2n-1} b_{2n}}{b_{2n-2}} A'_{n-1} - \frac{a_{2n-1} a_{2n-2} b_{2n}}{b_{2n-2}} A'_{n-2}$$

et une relation identique pour $B'_n \cdot A'_n$ et B'_n peuvent donc être considérés comme numérateur et dénominateur du $n^{\text{ième}}$ convergent d'une fraction continue dont les numérateurs et dénominateurs partiels ont la forme :

$$a'_n = - \frac{a_{2n-1} a_{2n-2} b_{2n}}{b_{2n-2}}$$

$$b'_n = \frac{a_{2n} b_{2n-2} + b_{2n} b_{2n-1} b_{2n-2} + a_{2n-1} b_{2n}}{b_{2n-2}}$$

Nous venons de réaliser la contraction de la fraction continue $b_0 + \frac{a_1}{b_1} + \dots$ ou

$$\text{fraction continue } b_0 + \frac{a_1 \quad b_2}{b_1 \quad b_2 + a_2} - \frac{a_2 \quad a_3 \quad b_4}{(b_2 b_3 + a_3) \quad b_4 + b_2 \quad a_4} - \dots$$

$$\dots - \frac{a_{2n-2} a_{2n-1} b_{2n-4} b_{2n}}{(b_{2n-2} b_{2n-1} + a_{2n-1}) b_{2n} + b_{2n-2} a_{2n}}$$

Il est possible de transformer une série en fractions continues :

$$\begin{aligned}
 \sum_{i=0}^{\infty} c_i x^i &= c_0 + \cfrac{c_1 x}{1 - \cfrac{\cfrac{c_2}{c_1} x}{1 + \cfrac{c_2}{c_1} x}} \\
 &= \cfrac{c_0}{1} - \cfrac{\cfrac{c_1}{c_0} x}{1 + \cfrac{c_1}{c_0} x} - \dots \\
 &= c_0 + \cfrac{c_1 x}{1} - \cfrac{c_2 x}{c_1 + c_2 x} - \cfrac{c_1 c_3 x}{c_2 + c_3 x} - \dots \\
 &= c_0 - \cfrac{c_1 x}{c_0 + c_1 x} - \cfrac{c_0 c_2 x}{c_1 + c_2 x}
 \end{aligned}$$

A partir d'une série donnée on peut trouver une fraction continue dont le développement du n^{ième} convergent en puissances croissantes de x coïncide avec la série jusqu'au terme en $c_n x^n$. C'est la fraction continue correspondante. En contractant la fraction continue correspondante par la méthode précédente on obtient une fraction continue dont le développement du n^{ième} convergent en puissances croissantes de x coïncide avec la série initiale jusqu'au terme en $c_{2n} x^{2n}$. C'est la fraction continue associée.

Considérons maintenant les convergents successifs de la fraction continue :

$$x + \cfrac{a_1}{x} + \cfrac{a_2}{x} + \dots$$

Ce sont des fonctions rationnelles de x. On peut donc espérer qu'il existe une connection entre la théorie des fractions continues, la table de Padé et l' ϵ -algorithme. En fait on peut comprendre la signification théorique de l' ϵ -algorithme de la façon suivante (Wynn (71)):

Etant donné la série $\sum_{i=0}^{\infty} c_i z^{-(i+1)}$, il est possible de construire une fraction continue de la forme :

$$\cfrac{c_0}{z - \alpha_0} - \cfrac{\beta_0}{z - \alpha_1} - \dots - \cfrac{\beta_{k-2}}{z - \alpha_{k-1}} - \dots$$

telle que son $n^{\text{ième}}$ convergent K_n ait un développement en puissances inverses croissantes de z identique avec la série jusqu'au terme en z^{-2n+1} , c'est-à-dire que $K_n = \sum_{i=0}^{\infty} c_i z^{-(i+1)} + \sum_{i=2n}^{\infty} d_i z^{-(i+1)}$ avec, en général, $d_i \neq 0$. On dit que cette fraction continue est associée à la série. Il est également possible de construire une autre fraction continue de la forme :

$$\cfrac{c_0}{z} - \cfrac{q_1}{1} - \cfrac{e_1}{z} - \dots - \cfrac{q_k}{1} - \cfrac{e_k}{z} - \dots$$

dont les convergents :

$$C_{2n} = \cfrac{c_0}{z} - \dots - \cfrac{e_{n-1}}{z} - \cfrac{q_n}{1} \quad n = 0, 1, \dots$$

$$C_{2n+1} = \cfrac{c_0}{z} - \dots - \cfrac{q_n}{1} - \cfrac{e_n}{z}$$

ont des développements en puissances inverses croissantes de z identiques avec la série donnée jusqu'aux termes en z^{-2n+1} et z^{-2n} respectivement. C'est la fraction continue correspondante. On a :

$$C_{2n} = K_n \quad n = 0, 1, \dots$$

Si on applique l'ε-algorithme avec $\varepsilon_0^{(0)} = 0$, $\varepsilon_0^{(n)} = \sum_{i=0}^{n-1} c_i z^{-(i+1)}$

pour $n = 1, 2, \dots$, on montre que :

$$\varepsilon_{2k}^{(0)} = K_k = C_{2k} \qquad \varepsilon_{2k}^{(1)} = C_{2k-1}$$

$$\varepsilon_{2k}^{(n)} = \sum_{i=0}^{n-1} c_i z^{-(i+1)} + z^{-n} \left(\frac{c_n}{z} - \frac{q_1}{1} - \dots - \frac{e_k}{z} \right)$$

De plus, soient $e_k^{(n)}$ et $q_k^{(n)}$ les éléments de diverses fractions continues :

$$\frac{c_n}{z} - \frac{q_1^{(n)}}{1} - \frac{e_1^{(n)}}{z}$$

dont le développement en puissances inverses croissantes de z de chacune d'entre elles est identique à la série jusqu'au terme en z^{-k} . Ces éléments $e_k^{(n)}$ et $q_k^{(n)}$ sont obtenus en employant l'algorithme QD de Rutishauser :

$$e_0^{(n)} = 0 \qquad q_1^{(n)} = C_{n+1} / C_n$$

$$q_k^{(n)} + e_k^{(n)} = q_k^{(n+1)} + e_{k-1}^{(n+1)}$$

$$q_{k-1}^{(n)} e_k^{(n)} = q_k^{(n+1)} e_k^{(n+1)}$$

on montre que :

$$\varepsilon_{2k}^{(n)} = \sum_{i=0}^{n-1} c_i z^{-(i+1)} + z^{-n} \left(\frac{c_n}{z} - \frac{q_1^{(n)}}{1} - \frac{e_1^{(n)}}{z} - \dots - \frac{e_{k-1}^{(n)}}{z} - \frac{q_k^{(n)}}{1} \right)$$

$$\varepsilon_{2k}^{(n+1)} = \sum_{i=0}^{n-1} c_i z^{-(i+1)} + z^{-n} \left(\frac{c_n}{z} - \dots - \frac{q_k^{(n)}}{1} - \frac{e_k^{(n)}}{z} \right)$$

Si $\sum_{i=0}^{\infty} c_i z^{-(i+1)}$ est le développement en série d'une fonction rationnelle,

les quantités $q_k^{(n)}$ tendent, pour n tendant vers l'infini, vers les racines de son dénominateur. L' \mathcal{E} -algorithme apparaît ainsi comme une méthode pour transformer les sommes partielles d'une série en convergents successifs de ses fractions continues associée et correspondante.

Notons que les diverses questions associées au problème de l'obtention de fonctions rationnelles à partir de développements en séries peuvent-être unifiées en utilisant la théorie classique des polynômes orthogonaux. Pour s'en convaincre il n'y a qu'à considérer la fraction continue

$$\cfrac{1}{a_0x+b_0} - \cfrac{c_1}{a_1x+b_1} - \cfrac{c_2}{a_2x+b_2}$$

Le $n^{\text{ième}}$ convergent A_n / B_n s'arrête au terme $a_{n-1}x + b_{n-1}$. A_n et B_n satisfont nous l'avons vu, la même relation de récurrence :

$$X_{n+1} = (a_nx + b_n) X_n - C_n X_{n-1}$$

avec comme conditions initiales $X_0 = 0, X_1 = 1$ pour A_n et $X_0 = 0, X_1 = a_0x + b_0$ pour B_n . Cette relation est analogue aux relations de récurrence satisfaites par les polynômes orthogonaux. De plus, on voit, que $s_i X_{n-1}$ est un polynôme de degré $n-1$ et X_n un polynôme de degré n , alors X_{n+1} sera un polynôme de degré $n+1$. Ces diverses questions ont été mises en évidence par Wynn (71), qui a étudié en détail la connexion entre l' \mathcal{E} -algorithme et les fractions continues (74) et à tenté une généralisation des polynômes orthogonaux(56). Mais cela nous entrainerait trop loin d'en parler ici.

2-3-5 Généralisation aux cas non scalaires

L' \mathcal{E} -algorithme peut être généralisé aux cas où les itérés initiaux ne sont plus des scalaires mais des matrices ou des vecteurs. La généralisation au cas matriciel ne pose aucun problème car on sait définir l'inverse d'une matrice carré. On a alors l'algorithme suivant :

$$E_{k+1}^{(n)} = E_{k-1}^{(n+1)} + \left[E_k^{(n+1)} \quad - E_k^{(n)} \right]^{-1}$$

Par contre la généralisation au cas vectoriel pose le problème de l'inverse d'un vecteur. Wynn (59) utilise dans ce cas la définition de l'inverse d'un vecteur due à Samuelson :

$$y^{-1} = \frac{\overline{y}}{(y, y)}$$

où \overline{y} est le conjugué de y et (y, y) le produit scalaire. Avec cette définition y^{-1} est le point inverse de y par rapport à la sphère unité de \mathbb{C}^p . y^{-1} est également le transposé de l'inverse généralisé de y au sens de Moore et Penrose pour les matrices rectangles. On a donc l'algorithme :

$$\varepsilon_{k+1}^{(n)} = \varepsilon_{k-1}^{(n+1)} + \frac{\overline{\Delta \varepsilon_k^{(n)}}}{(\Delta \varepsilon_k^{(n)}, \Delta \varepsilon_k^{(n)})}$$

Wynn a formulé la conjecture suivante qui a été démontrée par McLeod (41) ;

Si on a entre les vecteurs $\varepsilon_0^{(n)}$ une relation de la forme :

$$\sum_{i=0}^k c_i \varepsilon_0^{(n+i)} = s \sum_{i=0}^k c_i \quad \forall n = 0, 1, \dots$$

alors $\varepsilon_{2k}^{(n)} = s \quad \forall n$

Cet algorithme présente donc la même propriété que l' ε -algorithme scalaire. La fonction $\Psi(n, a)$ correspondante est donc identique. Cependant l'emploi de cet algorithme est plus simple que l'emploi de p algorithmes scalaires. On peut envisager d'appliquer la formule précédente, non plus à des vecteurs, mais aux éléments d'un espace de Hilbert quelconque et, plus généralement comme l'a démontré McLeod (41), aux éléments d'une algèbre de division associative sur \mathbb{C} .

2-4 Le p -algorithme

Nous avons vu de quelle manière le fait de prendre comme fonction $\Psi(n, a)$ celle du théorème II.2 conduit à l' ε -algorithme.

Prenons maintenant une fonction $\varphi(x, a)$ de la forme :

$$\varphi(x, a) = \frac{\sum_{i=0}^{k-1} a_i x^i}{\sum_{i=0}^k a_{k+i} x^i}$$

on a $\lim_{x \rightarrow \infty} \varphi(x, a) = 0 \quad \forall a$ avec $a_{2k} \neq 0$.

Soit $V_\varphi \{y'_n\}$ la variété de E des y tels que :

$$\langle y, y'_\infty - y'_n \rangle = \varphi(x_n, a) \quad \forall n \in \mathbb{N}$$

Il est possible de calculer $\langle y, y'_\infty \rangle$, $\forall y \in V_\varphi \{y'_n\}$ à partir de certains $\langle y, y'_n \rangle$. On utilise pour cela les différences inverses (ou reciproques) et la formule d'interpolation de Thiele (voir par exemple (42 ou (43)):

Soit $f(x)$ une fonction connue pour $x = x_0, x_1, \dots$ et supposons les x_i tous différents. On appelle différence réciproque d'ordre 0 la quantité $\rho_0(x_k) = f(x_k)$

On définit de même :

$$\rho_1(x_i, x_j) = \frac{x_i - x_j}{\rho_0(x_i) - \rho_0(x_j)}$$

$$\rho_2(x_0, x_1, x_2) = \frac{x_0 - x_2}{\rho_1(x_0, x_1) - \rho_1(x_1, x_2)} + \rho_0(x_1)$$

et de façon générale :

$$\rho_n(x_k, \dots, x_{k+n}) = \frac{x_k - x_{k+n}}{\rho_{n-1}(x_k, \dots, x_{k+n-1}) - \rho_{n-1}(x_{k+1}, \dots, x_{k+n})} + \rho_{n-2}(x_{k+1}, \dots, x_{k+n-1})$$

Posons $\rho_i^{(j)} = \rho_i(x_j, \dots, x_{j+i})$. La relation précédente devient :

$$\rho_{i+1}^{(k)} = \rho_{i-1}^{(k+1)} + \frac{x_{k+i+1} - x_k}{\rho_i^{(k+1)} - \rho_i^{(k)}} \quad (II-7)$$

A partir des différences réciproques on peut développer $f(x)$ en fraction continue à l'aide du système :

$$f(x) = \rho_0^{(0)} + \frac{x - x_0}{\rho_1(x, x_0)}$$

$$\rho_1(x, x_0) = \rho_1^{(0)} + \frac{x - x_1}{\rho_2(x, x_0, x_1) - \rho_0^{(0)}}$$

.....

$$\rho_{n-1}(x, x_0, \dots, x_{n-2}) = \rho_{n-1}^{(0)} + \frac{x - x_{n-1}}{\rho_n(x, x_0, \dots, x_{n-1}) - \rho_{n-2}^{(0)}}$$

d'où

$$f(x) = \rho_0^{(0)} + \frac{x - x_0}{\rho_1^{(0)}} + \frac{x - x_1}{\rho_2^{(0)} - \rho_0^{(0)}} + \frac{x - x_2}{\rho_3^{(0)} - \rho_1^{(0)}} + \dots$$

ou encore

$$f(x) = \alpha_0 + \frac{x - x_0}{\alpha_1} + \frac{x - x_1}{\alpha_2} + \dots$$

avec

$$\alpha_0 = \rho_0^{(0)}, \alpha_1 = \rho_1^{(0)} \quad \text{et} \quad \alpha_k = \rho_k^{(0)} - \rho_{k-2}^{(0)} \quad \text{pour } k > 1$$

On calcule les convergents successifs de cette fraction continue à l'aide des relations habituelles de récurrence et on montre par induction que $C_n(x) = P_n(x) / Q_n(x)$ avec :

$$P_{2k}(x) = A_0 + A_1x + \dots + A_{k-1}x^{k-1} + x^k$$

$$Q_{2k}(x) = B_0 + B_1x + \dots + B_{k-2}x^{k-2} + \rho_{2k-1}^{(0)} x^{k-1}$$

$$P_{2k+1}(x) = C_0 + C_1x + \dots + C_{k-1}x^{k-1} + \rho_{2k}^{(0)} x^k$$

$$Q_{2k+1}(x) = D_0 + D_1x + \dots + D_{k-1}x^{k-1} + x^k$$

Donc, si on termine la fraction continue à α_{2k-1} on obtient $f(x) = P_{2k}(x) / Q_{2k}(x)$ et si on la termine à α_{2k} on a $f(x) = P_{2k+1}(x) / Q_{2k+1}(x)$. Dans ce second cas, so l'on fait tendre x vers l'infini, on a :

$$\lim_{x \rightarrow \infty} f(x) = \rho_{2k}^{(0)}$$

on voit que si $y \in V_\varphi \{y'_n\}$ alors $\langle y, y'_n \rangle$ sera de la forme :

$$\frac{C_0 + C_1x_n + \dots + C_{k-1}x_n^{k-1} + \rho_{2k}^{(0)} x_n^k}{D_0 + D_1x_n + \dots + D_{k-1}x_n^{k-1} + x_n^k}$$

et que, par conséquent, on aura $\langle y, y'_\infty \rangle = \rho_{2k}^{(0)}$

Si l'on prend $x_k = k$ on obtient à partir de la relation (II-7) la forme habituelle du ρ -algorithme qui a été donnée par Wynn (65) :

$$\rho_{i+1}^{(k)} = \rho_{i-1}^{(k+1)} + \frac{i+1}{\rho_i^{(k+1)} - \rho_i^{(k)}} \quad (\text{II-8})$$

avec

$$\rho_{-1}^{(k)} = 0 \quad \text{et} \quad \rho_0^{(k)} = \langle y, y'_k \rangle$$

on voit ~~que~~ la similitude de cette relation avec la relation (II-5) de l' ε -algorithme. Les quantités calculées par (II-8) sont rangées dans ~~une~~ ^{une} table identique à celle utilisée pour les $\varepsilon_k^{(n)}$. Ici encore l'indice inférieur représente le numéro de la colonne et l'indice supérieur celui de la diagonale.

Seules les colonnes paires sont utiles, les colonnes impaires sont des calculs intermédiaires. Le φ -algorithme appartient à la classe des algorithmes de losange. Quand $\varphi_i^{(k)}$ et $\varphi_i^{(k+1)}$ deviennent égaux ou très voisins on utilise à la place de (II-8) des règles particulières analogues à celles employées pour l' \mathcal{E} -algorithme. Nous les verrons au chapitre V.

2-4-1 Forme étendue du φ -algorithme

La forme étendue du φ -algorithme est donnée par la formule (II-7). Elle est plus générale que la forme (II-8) proposée par Wynn car elle permet de rendre compte d'une dépendance par rapport à un paramètre x plus générale qu'une simple numérotation. En effet, la fonction $\varphi(x, a)$ correspondant à (II-7) est :

$$\varphi(x, a) = \frac{\sum_{i=0}^{k-1} a_i x^i}{\sum_{i=0}^k a_{k+i} x^i}$$

tandis que celle correspondant à (II-8) est $\varphi(n, a) = \frac{\sum_{i=0}^{k-1} a_i n^i}{\sum_{i=0}^k a_i n^i}$

Si les x_k sont équidistants on a $x_k = x_0 + kh$ $k= 0, 1, \dots$ et la forme (II-7) se réduit à la forme (II-8). Par contre si les x_k sont en progression géométrique on a :

$$x_k = a x_{k-1} = a^k x_0$$

d'où

$$\varphi_{i+1}^{(k)} = \varphi_{i-1}^{(k+1)} + \frac{a^k (a^{i+1} - 1)}{\varphi_i^{(k+1)} - \varphi_i^{(k)}}$$

2-4-2 Le φ^x -algorithme

Dans la fonction $\varphi(x, a)$ correspondant à la forme étendue (II-7) du

ρ -algorithme remplaçons la variable x par la variable x^r . Nous obtenons :

$$\varphi(x, a) = \frac{\sum_{i=0}^{k-1} a_i x^{ri}}{\sum_{i=0}^k a_{k+i} x^{ri}} = \sum_{i=1}^{\infty} b_i x^{-ri}$$

Soit $V_{\varphi}\{y'_n\}$ la variété correspondante. On a :

$$\langle y, y'_{\infty} \rangle = \rho_{2k}^{(0)} \quad \forall y \in V_{\varphi}\{y'_n\}$$

Les $\rho_i^{(k)}$ sont calculés à l'aide du ρ^r -algorithme (II) :

$$\rho_{i+1}^{(k)} = \rho_{i-1}^{(k+1)} + \frac{x_{k+i+1}^r - x_k^r}{\rho_i^{(k+1)} - \rho_i^{(k)}} \quad (\text{II-9})$$

avec $\rho_{-1}^{(k)} = 0$ et $\rho_0^{(k)} = \langle y, y'_k \rangle$

Cette forme (II-9) est encore plus générale que (II-7). Elle permet de rendre compte d'une dépendance par rapport à x qui ne comporte que des puissances entières de x^{-r} . Par exemple, quand on intègre une fonction par la méthode des trapèzes, on sait que l'erreur admet un développement en série ne contenant que des puissances paires du pas. On prendra dans ce cas $r = 2$. Si, de plus, on choisit des pas en progression géométrique de raison $1/2$, (II-9) devient :

$$\rho_{i+1}^{(k)} = \rho_{i-1}^{(k+1)} + \frac{2^{2k} (2^{2i+2} - 1)}{\rho_i^{(k+1)} - \rho_i^{(k)}}$$

Cet algorithme d'extrapolation est à rapprocher de la méthode de Romberg. Les résultats numériques obtenus avec ce ρ^2 -algorithme sont souvent meilleurs comme nous le verrons au dernier chapitre.

2-4-3 Propriétés du ρ -algorithme

Le ρ -algorithme est un algorithme non linéaire au même titre que l' ε -algorithme. Comme lui il est linéaire par rapport à la multiplication par un scalaire et par rapport à l'addition d'un scalaire. Si partant de $\rho_0^{(n)}$ on obtient $\rho_k^{(n)}$ alors, partant de $\overline{\rho}_0^{(n)} = a \rho_0^{(n)} + b$, on obtiendra $\overline{\rho}_{2k}^{(n)} = a \rho_{2k}^{(n)} + b$ et $\overline{\rho}_{2k+1}^{(n)} = \rho_{2k+1}^{(n)} / b$ ce qui peut se mettre sous la forme symbolique :

$$\rho_{2k}^{(n)} (a f(x) + b) = a \rho_{2k}^{(n)} (f(x)) + b$$

$$\text{et } \rho_{2k+1}^{(n)} (a f(x) + b) = \rho_{2k+1}^{(n)} (f(x)) / b$$

on a de même :

$$\rho_{2k}^{(n)} (1 / f(x)) = 1 / \rho_{2k}^{(n)} (f(x))$$

et plus généralement :

$$\rho_{2k}^{(n)} \left(\frac{a f(x) + b}{c f(x) + d} \right) = \frac{a \rho_{2k}^{(n)} (f(x)) + b}{c \rho_{2k}^{(n)} (f(x)) + d}$$

on a également la propriété :

$$\varepsilon_2^{(n)} = \frac{1}{2} \left[\rho_2^{(n)} + \langle y, y'_{n+1} \rangle \right]$$

2-5 Les Algorithmes de losange

On a vu se dégager, tout au long de cet exposé, un certain nombre de ressemblances entre les algorithmes étudiés. Ces algorithmes relient entre elles des séries, des fonctions rationnelles et des fractions continues. Ils font généralement appel à la table de Padé et aux polynômes orthogonaux. Enfin ils appartiennent à la classe des algorithmes de losange c'est-à-dire des algorithmes

qui relie des quantités situées aux quatre angles d'un losange dans une table analogue à la table \mathcal{E} .

Bauer (5,6) a mis en évidence la relation qui existe entre certains de ces algorithmes. Il a trouvé un autre algorithme de losange, le g -algorithme qui semble être l'algorithme de base. De cet algorithme on peut déduire les algorithmes QD et \mathcal{E} ainsi que l' η -algorithme qui leur sert de lien.

2-5-1 La décomposition g

On obtient le g -algorithme à partir de la décomposition g d'une fraction continue du type de Stieltjes (ou S -fraction). Une S -fraction est une fraction continue de la forme :

$$s(z) = \cfrac{e_0}{z} - \cfrac{q_1}{1} - \cfrac{e_1}{z} - \cfrac{q_2}{1} - \dots$$

où les scalaires q_i et e_i sont des réels négatifs. Les dénominateurs des convergents successifs de cette S -fraction sont des polynômes $z \cdot p_0^*(z)$, $p_1(z)$, $z \cdot p_1^*(z)$, $p_2(z)$, ..., $p_k(z)$ et $p_k^*(z)$ sont des polynômes de degré k avec le coefficient du terme en z^k égal à un. Ils obéissent aux relations de récurrence.

$$\left. \begin{aligned} p_0(z) &= p_0^*(z) = 1 \\ p_k(z) &= z p_{k-1}^*(z) - q_k p_{k-1}(z) \\ p_k^*(z) &= p_k(z) - e_k p_{k-1}^*(z) \end{aligned} \right\} \quad k = 1, 2, \dots$$

La suite $p_0^*(z)$, $p_1(z)$, $p_1^*(z)$, étant donnée, on obtient, identiquement par rapport à la variable c :

$$q_k = \frac{p_{k-1}^*(c)}{p_{k-1}(c)} \left(c - \frac{p_k(c)}{p_{k-1}^*(c)} \right)$$

$$e_k = \frac{p_k(c)}{p_{k-1}^*(c)} \left(1 - \left(\frac{p_k^*(c)}{p_k(c)} \right) \right)$$

Introduisons les quantités $g_k = g_k(c)$ définies par :

$$g_{2k} = \frac{p_k^*(c)}{p_k(c)} \quad g_{2k+1} = \frac{p_{k+1}(c)}{p_k^*(c)} \quad k = 0, 1, \dots$$

on obtient la décomposition g de la S-fraction :

$$S(z) = \cfrac{e_0}{z} - \cfrac{g_0(c-g_1)}{1} - \cfrac{g_1(1-g_2)}{z} - \cfrac{g_2(c-g_3)}{1} - \dots$$

soit $R(z) = \frac{S_0}{z} + \frac{S_1}{z^2} + \dots$ le développement en série de puissances qui coïncide avec les n premiers termes du développement en puissances inverses croissantes du $n^{\text{ième}}$ convergent de la S-fraction. On obtient, après un certain nombre de transformations :

$$z s(z) = s_0 + \cfrac{s_0(c-g_1)}{z-c} + \cfrac{g_1 g_2}{1} + \cfrac{(1-g_2)(c-g_3)}{z-c} + \cfrac{g_3 g_4}{1} + \dots$$

introduisons la quantité $s^{(1)}(z) = z \cdot s(z) - s_0$, on a :

$$s^{(1)}(z) = \cfrac{s_0}{z-c} + \cfrac{g_0^{(1)} g_1^{(1)}}{1} + \cfrac{(c-g_1^{(1)})(1-g_2^{(1)})}{z-c} + \dots$$

on répète ensuite le processus avec $s^{(n)}(z) =$ partie principale de $z^n \cdot s(z) \approx \frac{s_n}{z} + \frac{s_{n+1}}{z^2} + \dots$. Les quantités $g_k^{(n)}$ peuvent être placées dans un tableau

à double entrée analogue aux tableaux \mathcal{P} et \mathcal{E} . De plus les quantités de la tables sont liées par un algorithme de losange :

$$g_{2k-1}^{(n-1)} \cdot g_{2k}^{(n-1)} = g_{2k-2}^{(n)} \cdot g_{2k-1}^{(n)} \quad k = 1, 2, \dots$$

$$\left[1-g_{2k}^{(n-1)} \right] \cdot \left[c-g_{2k-1}^{(n-1)} \right] = \left[c-g_{2k-1}^{(n)} \right] \cdot \left[1-g_{2k}^{(n)} \right] \quad n = 0, 1, \dots$$

et de plus $s_k = s_{k-1} \cdot (c - g_1^{(k-1)})$ on a :

$$s^{(n)}(z) = \frac{s_n}{z - c} + \frac{g_0^{(n)} g_1^{(n)}}{1} + \frac{(c - g_1^{(n)})(1 - g_2^{(n)})}{z - c} + \dots$$

Si l'on pose :

$$q_k^{(n)} = g_{2k-2}^{(n)} \cdot (c - g_{2k-1}^{(n)})$$

$$e_k^{(n)} = g_{2k-1}^{(n)} \cdot (1 - g_{2k}^{(n)})$$

on retrouve les relations de l'algorithme QD et on a :

$$s^{(n)}(z) = \frac{s_n}{z} - \frac{q_1^{(n)}}{1} - \frac{e_1^{(n)}}{z} - \frac{q_2^{(n)}}{1} - \dots$$

Si l'on pose :

$$\eta_k^{(n)} = s_n \cdot \frac{c - g_1^{(n)}}{g_1^{(n)}} \cdot \frac{1 - g_2^{(n)}}{g_2^{(n)}} \dots \left\{ \begin{array}{l} \frac{1 - g_k^{(n)}}{g_k^{(n)}} \text{ si } k \text{ est pair} \\ \frac{c - g_k^{(n)}}{g_k^{(n)}} \text{ si } k \text{ est impair} \end{array} \right.$$

On obtient :

$$c \left[\eta_{2k-1}^{(n)} + \eta_{2k}^{(n)} \right] = \eta_{2k-2}^{(n+1)} + \eta_{2k-1}^{(n+1)}$$

$$\frac{1}{c} \cdot \left[\frac{-\frac{1}{\eta_{2k}^{(n)}}}{\eta_{2k}^{(n)}} + \frac{\frac{1}{\eta_{2k+1}^{(n)}}}{\eta_{2k+1}^{(n)}} \right] = \frac{1}{\eta_{2k-1}^{(n+1)}} + \frac{1}{\eta_{2k}^{(n+1)}}$$

avec $\frac{1}{\eta_{-1}^{(n)}} = 0$. On a :

$$s^{(n)}(c) = \frac{1}{c} \sum_{i=0}^{\infty} \eta_i^{(n)}$$

Si l'on pose :

$$\varepsilon_{2k}^{(n)} = \sum_{i=0}^{n-1} \frac{\eta_0^{(i)}}{c^{i+1}} + \frac{1}{c^{n+1}} \cdot \sum_{i=0}^{2k-1} \eta_i^{(n)} \quad \varepsilon_0^{(0)} = 0$$

$$\varepsilon_{2k+1}^{(n)} = c^{n+1} \cdot \sum_{i=0}^{2k} \frac{1}{\eta_i^{(n)}} \quad \varepsilon_{-1}^{(n)} = 0$$

on retrouve l' ε -algorithme : $(\varepsilon_{k+1}^{(n+1)} - \varepsilon_k^{(n)}) \cdot (\varepsilon_{k+1}^{(n)} - \varepsilon_{k-1}^{(n+1)}) = 1$

$\varepsilon_{2k}^{(0)}$ est le $2k$ ^{ième} convergent de la S-fraction. Il a comme dénominateur un polynôme de degré k . Son développement en série coïncide avec les $2k$ premiers termes de $R(c)$. De plus, si $\lambda_1, \dots, \lambda_k$ sont les racines du dénominateur du $2k$ ^{ième} convergent, on a :
$$\varepsilon_{2k}^{(0)} = \sum_{i=1}^k \frac{e_i}{c - \lambda_i}$$

2-5-2 Equations aux dérivées partielles associées aux algorithmes de losange

On peut considérer (69) les algorithmes de losange comme des approximations aux différences du premier ordre de systèmes d'équations aux dérivées partielles du premier ordre par rapport à deux variables indépendantes. Les algorithmes de losange sont de la forme :

$$\theta_k^{(n)} \left[\phi_{k+1}^{(n)}, \phi_k^{(n)}, \phi_k^{(n+1)}, \phi_{k-1}^{(n+1)} \right] = 0 \quad (\text{II-10})$$

avec $\phi_{-1}^{(n)}$ et $\phi_0^{(n)}$ donnés pour $n = 0, 1, \dots$

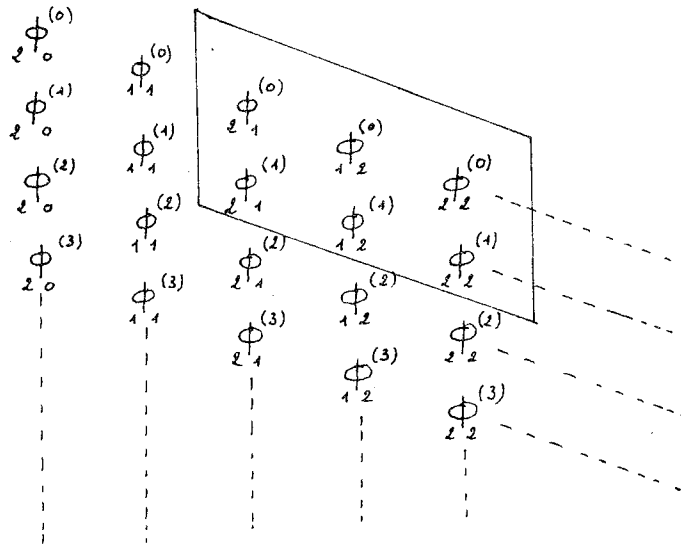
Nous avons vu que les quantités d'indice inférieur pair ne jouent pas le même rôle que les quantités d'indice inférieur impair. Nous les distinguerons en posant :

$$\phi_{1 \ k}^{(n)} = \phi_{2k-2}^{(n)} \quad \phi_{2 \ k}^{(n)} = \phi_{2k-1}^{(n)}$$

on remplace alors la relation (II-10) de l'algorithme par les deux relations :

$$\begin{aligned} \theta_k^{(n)} \left[\phi_{2 \ k-1}^{(n)}, \phi_{1 \ k}^{(n)}, \phi_{2 \ k-1}^{(n+1)}, \phi_{1 \ k-1}^{(n+1)} \right] &= 0 \\ \theta_k^{(n)} \left[\phi_{1 \ k}^{(n)}, \phi_{2 \ k}^{(n)}, \phi_{1 \ k}^{(n+1)}, \phi_{2 \ k-1}^{(n+1)} \right] &= 0 \end{aligned} \quad (\text{II-11})$$

C'est l'algorithme dual de losange. Les quantités $\phi_{1\ k}^{(n)}$ et $\phi_{2\ k}^{(n)}$ se plaçant dans un tableau à deux dimensions :



Les quantités contenues dans le losange en traits pleins sont celles utilisées dans les relations (II-11) avec $n = 0$ et $k = 2$.

Considérons k et n comme les coordonnées du point où se trouve la quantité $\phi_k^{(n)}$ dans le tableau ϕ .

Introduisons, à la place de k et n , les coordonnées continues $x = a + 2kh$ et $y = b + 2(n+k).h$, où a et b sont des constantes et h le pas.

Le système de coordonnées précédent ;

$$\begin{array}{ccccc}
 & & 1, k, n-1 & & \\
 & & & & \\
 & & 2, k-1, n & & 2, k, n-1 \\
 & & & & \\
 1, k-1, n+1 & & 1, k, n & & 1, k+1, n-1 \\
 & & & & \\
 & & 2, k-1, n+1 & & 2, k, n \\
 & & & & \\
 & & & & 1, k, n+1
 \end{array}$$

est remplacé par :

$$\begin{array}{ccccc}
 & & & & x, y-2h \\
 & & & & \\
 & & x-h, y-h & & x+h, y-h \\
 & & & & \\
 x-2h, y & & x, y & & x+2h, y \\
 & & & & \\
 & & x-h, y+h & & x+h, y+h \\
 & & & & \\
 & & & & x, y+2h
 \end{array}$$

Des relations (II-11) de l'algorithme dual de losange on dérive les équations aux dérivées partielles en remplaçant respectivement ${}_1\phi_k^{(n)}$ et ${}_2\phi_k^{(n)}$ par ${}_1\phi(x,y)$ et ${}_2\phi(x+h,y+h)$ et en faisant tendre h vers zéro.

Prenons comme exemple un algorithme de losange qui serait donné par :

$$\phi_{k-1}^{(n)} + \phi_k^{(n)} - \phi_{k-1}^{(n+1)} - \phi_{k-2}^{(n+1)} = 0 \quad (II-12)$$

on obtient pour l'algorithme dual :

$$\begin{cases} {}_1\phi_k^{(n)} - {}_1\phi_{k-1}^{(n+1)} = {}_2\phi_{k-1}^{(n+1)} - {}_2\phi_{k-1}^{(n)} \\ {}_2\phi_k^{(n)} - {}_2\phi_{k-1}^{(n+1)} = {}_1\phi_k^{(n+1)} - {}_1\phi_k^{(n)} \end{cases}$$

et après quelques manipulations :

$${}_1\phi_{k+1}^{(n-1)} - {}_1\phi_k^{(n-1)} + {}_1\phi_{k-1}^{(n+1)} - {}_1\phi_k^{(n+1)} = 0$$

et une relation identique pour ${}_2\phi_k^{(n)}$

En passant des variables discrètes aux variables continues et en faisant tendre h vers zéro on trouve ${}_1\phi'_x = {}_2\phi'_y$ et ${}_1\phi'_y = {}_2\phi'_x$ ou encore

$${}_1\phi''_{xx} - {}_1\phi''_{yy} = 0$$

qui est l'équation aux dérivées partielles dont (II-12) est tirée par approximation aux différences finies du premier ordre.

Pour les algorithmes de losange que nous connaissons on obtient de même :

1°) - \mathcal{E} -algorithme

Faisons les substitutions $\mathcal{E}_{2k-1}^{(n)} = h^{-1} \cdot {}_1\mathcal{E}(x,y)$ et $\mathcal{E}_{2k}^{(n)} = h^{-1} \cdot {}_2\mathcal{E}(x,y)$

on obtient : ${}_1\mathcal{E}'_x \cdot {}_2\mathcal{E}'_y = 1$ et ${}_2\mathcal{E}'_x \cdot {}_1\mathcal{E}'_y = 1$. ${}_1\mathcal{E}(x,y)$ et ${}_2\mathcal{E}(x,y)$ vérifient chacun la relation :

$$\left(\frac{1}{\mathcal{E}'_y} \right)_y = \left(\frac{1}{\mathcal{E}'_x} \right)_x$$

C'est l'équation aux dérivées partielles de la surface de Padé. Il existe de plus entre ${}_1\varepsilon(x,y)$ et ${}_2\varepsilon(x,y)$ une relation de la forme $f({}_1\varepsilon, {}_2\varepsilon) = 0$.

2°) - Algorithme QD

on obtient $q'_y = e'_x$, $e'_x = q'_y$ et

$$\left(\frac{q''_{yy} - ((\log q)'_x \cdot q'_y)}{(\log q)''_{xx}} \right)'_x = q'_y$$

et une relation analogue en remplaçant q par e et en intervertissant x et y .

3°) - q-algorithme

avec les substitutions $g_{2k-1}^{(n)} = {}_1g(x,y)$ et $g_{2k}^{(n)} = {}_2g(x,y)$ on obtient :

$${}_2g \cdot {}_1g'_y = {}_1g \cdot {}_2g'_x \quad \text{et} \quad (1-{}_2g) \cdot {}_1g'_x = (c - {}_1g) \cdot {}_2g'_x$$

4°) - η -algorithme

on trouve ${}_2\eta'_x = {}_1\eta'_y$ et $\left(\frac{1}{{}_1\eta'_x}\right)' = \left(\frac{1}{{}_2\eta'_y}\right)'$. On a également :

$${}_2\varepsilon'_y = {}_2\eta, \quad {}_2\varepsilon'_x = {}_1\eta, \quad {}_1\varepsilon'_y = {}_1\eta^{-1} \quad \text{et} \quad {}_1\varepsilon'_x = {}_2\eta^{-1}.$$

5°) - ρ -algorithme

Avec les substitutions $h^2 \rho_{2k-1}^{(n)} = {}_1\rho(x,y)$ et $h \rho_{2k}^{(n)} = {}_2\rho(x,y)$ on obtient :

$${}_1\rho'_x \cdot {}_2\rho'_y = x^{-a} \quad \text{et} \quad {}_1\rho'_y \cdot {}_2\rho'_x = x^{-a}.$$

${}_1\rho(x,y)$ et ${}_2\rho(x,y)$ vérifient chacun la relation :

$$\left(\frac{x-a}{\rho'_y} \right)'_y = \left(\frac{x-a}{\rho'_x} \right)'_x$$

Il existe de plus entre ${}_1\rho(x,y)$ et ${}_2\rho(x,y)$ une relation de la forme $f({}_1\rho, {}_2\rho) = 0$

L'intérêt des équations aux dérivées partielles associées aux algorithmes de losange apparait dans l'étude des conditions de stabilité de ces algorithmes comme nous le verrons au chapitre V.

2-6 Accélération des suites hypergéométriques

L' ϵ -algorithme n'est capable d'accélérer que des suites d'ordre 1. Après avoir donné la définition de l'ordre de convergence d'une suite nous allons construire un algorithme capable d'accélérer les suites d'ordre supérieur à 1. L'intérêt de cet algorithme réside surtout dans l'exemple de construction d'une méthode d'extrapolation basée sur une fonction $\Psi(n,a)$ donnée.

Définition II-1 Ordre de convergence d'une suite (18)

Soit E un espace métrique et soit $\{x_n\}$ une suite d'éléments de E qui converge vers x_0 . On dit que l'ordre de convergence de la suite $\{x_n\}$ est $k \geq 1$ si $\exists K > 0$ tel que :

$$d(x_{n+1}, x_\infty) \leq K [d(x_n, x_\infty)]^k \quad \forall n$$

Nous voulons maintenant construire un algorithme capable d'accélérer les suites d'ordre $k \neq 1$. Prenons $E = \mathbb{R}$ et supposons que :

$$d_{n+1} = K d_n^k \quad \text{avec} \quad d_n = x_n - x_\infty$$

on a :

$$d_1 = K d_0^k$$

$$d_2 = K d_1^k = K^{k+1} d_0^{k^2}$$

$$d_n = K^{\frac{n-1}{k} + \frac{n-2}{k^2} + \dots + k+1} d_0^{k^n}$$

si $k = 1$ $d_n = K^n d_0 = D e^n$

Si $k \neq 1$

$$d_n = K^{\frac{n}{k-1}} d_0^{k^n} = K^{\frac{1}{k-1}} \left[K^{\frac{1}{k-1}} d_0 \right]^{k^n} = D e^{k^n}$$

avec $e = K$ et $D = d_0$ pour $k = 1$ et $e = K^{\frac{1}{k-1}}$ d_0 et $D = K^{\frac{1}{k-1}}$ pour $k \neq 1$.

Il est facile de montrer que si $k > 1$, l' ϵ -algorithme n'est capable ni

d'accélérer ni d'améliorer la convergence. On a en effet, dans ce cas :

$$\lim_{n \rightarrow \infty} \frac{\varepsilon_{2k}^{(n)} - x_{\infty}}{x_n - x_{\infty}} = 1 \quad \forall k$$

Soit donc une suite de la forme $x_n = x_{\infty} + D e^{k^n}$ avec $0 < e < 1$ et $k \neq 1$. La fonction $\varphi(n, a)$ correspondante est $\varphi(n, a) = a_1 \cdot a_2^{a_3}$ avec

$a_1 = D$, $a_2 = e$ et $a_3 = k$. Pour déterminer x_{∞} nous avons besoin de 4 itérés successifs puisqu'il y a 4 inconnues. Posons $b_n = e^{k^n}$, nous avons :

$$\begin{aligned} x_n &= x_{\infty} + D b_n & \Delta x_n &= D b_n (b_n^{k-1} - 1) \\ x_{n+1} &= x_{\infty} + D b_n^k & \Delta x_{n+1} &= D b_n^k (b_n^{k(k-1)} - 1) \\ x_{n+2} &= x_{\infty} + D b_n^{k^2} & \Delta x_{n+2} &= D b_n^{k^2} (b_n^{k^2(k-1)} - 1) \\ x_{n+3} &= x_{\infty} + D b_n^{k^3} \end{aligned}$$

d'où
$$D b_n = \frac{\Delta x_n}{b_n^{k-1} - 1} \quad (\text{II-13})$$

Il nous faut maintenant déterminer b_n^{k-1} . Posons $c_n = b_n^{k-1}$, $A_n = \frac{\Delta x_{n+1}}{\Delta x_n}$;

Nous avons à résoudre un système non linéaire de 2 équations à 2 inconnues :

$$\left. \begin{aligned} A_n &= c_n \frac{c_n^k - 1}{c_n - 1} \\ A_{n+1} &= c_n^k \frac{c_n^{k^2} - 1}{c_n^k - 1} \end{aligned} \right\} \quad (\text{II-14})$$

L'étude de ce système montre qu'il existe une et une seule solution c_n comprise

entre $\frac{A_n (A_{n+1} + 1)}{A_n (A_{n+1}) + 1}$ et $\text{Min} (1, A_n)$. On obtient cette valeur de c_n

en résolvant (II-14) par une méthode itérative quelconque, d'où $D b_n$ par (II-13) et $x_\infty = x_n - D b_n$.

Soit maintenant une suite telle que :

$$d(x_{n+1}, x_\infty) = \sum_{k>0} a_k [d(x_n, x_\infty)]^k$$

un nombre fini ou infini de coefficients a_k pouvant être nuls. Overholt (45) a proposé une méthode d'accélération de la convergence pour de telles suites. C'est une extension du procédé Δ^2 d'Aitken. Prenons $E = \mathbb{R}$, posons $d_n = d(x_n, x_\infty)$ et supposons que $x_0 = x_\infty + d_0$ et que $x_1 = x_\infty + a_1 d_0 + a_2 d_0^2 + \dots$. On obtient :

$$\frac{x_1 - a_1 x_0}{1 - a_1} = x_\infty + \frac{a_2 d_0^2}{1 - a_1} + \dots$$

En général a_1 est inconnu. On en détermine une approximation du premier ordre en utilisant x_2 :

$$x_2 = x_\infty + a_1 d_1 + a_2 d_1^2 + \dots$$

ce qui donne :

$$\bar{a}_1 = \frac{x_2 - x_1}{x_1 - x_0} = a_1 + (1 + a_1) a_2 d_0 + \dots$$

cette approximation est suffisante pour que la quantité :

$$x_2^{(1)} = \frac{x_1 - \bar{a}_1 x_0}{1 - \bar{a}_1} = x_1 - \frac{\Delta x_0 \cdot \Delta x_1}{\Delta^2 x_0} = \varepsilon_2^{(0)}$$

soit une approximation du second ordre de x_∞ :

$$x_2^{(1)} = x_\infty - \frac{a_1 a_2}{1 - a_1} d_0^2 + \dots$$

De même toute approximation \bar{a}_1 du premier ordre conduit à une formule du second ordre. Cette remarque est à la base de l'extension du procédé d'Aitken

proposée par Overholt. Cette extension est analogue à celle qui permet, dans la méthode de Richardsen, de passer d'un polynôme de degré k à un polynôme de degré $k+1$

Afin d'uniformiser les notations écrivons $x_1^{(0)}$, $x_1^{(1)}$, $x_1^{(2)}$, ... au lieu de x_0 , x_1 , x_2 , ... supposons que :

$$d_1 = a_{11} d_0 + a_{21} d_0^2 + \dots \quad (\text{II-15})$$

avec $|a_{11}| < 1$ et $a_{11} \neq 0$. On utilise les $k+1$ premiers termes de cette suite comme arguments d'une fonction f_k pour obtenir une approximation d'ordre k de x_∞ :

$$\begin{aligned} x_k^{(1)} &= f_k(x_1^{(0)}, x_1^{(1)}, \dots, x_1^{(k)}) \\ &= x_\infty + a_{kk} d_0^k + a_{k+1,k} d_0^{k+1} + \dots \end{aligned}$$

Supposons à présent que $a_{kk} \neq 0$. L'indice supérieur 1 signifie que $x_k^{(1)}$ est le premier terme d'une suite d'approximations d'ordre k de x_∞ . La valeur suivante est :

$$\begin{aligned} x_k^{(2)} &= f_k(x_1^{(1)}, \dots, x_1^{(k+1)}) \\ &= x_\infty + a_{kk} d_1^k + a_{k+1,k} d_1^{k+1} + \dots \end{aligned}$$

et en général :

$$x_k^{(i)} = f_k(x_1^{(i-1)}, \dots, x_1^{(k+i-1)})$$

A partir de (II-15) on obtient :

$$d_1^k = a_1^k \cdot d_0^k + k a_1^{k-1} \cdot a_2 d_0^{k+1} + \dots$$

Ce qui donne :

$$x_k^{(2)} = x_\infty + a_{kk} a_1^k d_0^k + (a_{kk} k a_1^{k-1} a_2 + a_{k+1,k} a_1^{k+1}) d_0^{k+1} + \dots$$

Supposons connue une approximation du premier ordre \bar{a}_1 de a_1 :

$$\bar{a}_1 = a_1 + \alpha_1 d_0 + \alpha_2 d_0^2 + \dots$$

on obtient :

$$\bar{a}_1^{-k} = a_1^k + k a_1^{k-1} \alpha_1 d_0 + \dots$$

ce qui donne :

$$\frac{\bar{a}_1^{-k}}{1 - \bar{a}_1^{-k}} = \frac{a_1^k}{1 - a_1^k} + B d_0 + \dots \text{ avec } B = \frac{k a_1^{k-1} \alpha_1}{(1 - a_1^k)^2}$$

d'où finalement :

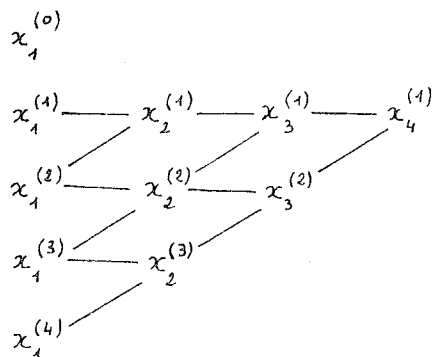
$$\begin{aligned} x_{k+1}^{(1)} &= f_{k+1} (x_1^{(0)}, \dots, x_1^{(k+1)}) \\ &= x_k^{(2)} + \frac{\bar{a}_1^{-k}}{1 - \bar{a}_1^{-k}} (x_k^{(2)} - x_k^{(1)}) \\ &= x_\infty + a_{kk} a_1^k d_0^k + \dots + \left(\frac{a_1^k}{1 - a_1^k} + B d_0 + \dots \right) (a_{kk} (a_1^k - 1) d_0^k + \dots) \\ &= x_\infty + a_{k+1,k+1} d_0^{k+1} + a_{k+2,k+1} d_0^{k+2} + \dots \end{aligned}$$

avec

$$a_{k+1,k+1} = \frac{a_1^{k-1}}{1 - a_1^k} \left(k (a_2 - \alpha_1) a_{kk} + (a_1^2 - a_1) a_{k+1,k} \right)$$

C'est-à-dire qu'à partir d'approximations d'ordre k nous avons généré une approximation d'ordre k+1. On peut donc bâtir à partir des valeurs initiales $x_1^{(0)}, x_1^{(1)}, \dots$

un tableau de la forme ;



Dans ce schéma la première colonne est identique à la seconde colonne du tableau

ε puisqu'elle correspond au procédé habituel d'Aitken : $\varepsilon_2^{(n-1)} = x_2^{(n)}$ $n = 1, 2, \dots$

pour le calcul de cette colonne on utilise comme approximation de a_1 :

$$\bar{a}_1 = \frac{x_1^{(i)} - x_1^{(i-1)}}{x_1^{(i-1)} - x_1^{(i-2)}}$$

D'où comme algorithme général :

$$x_{k+1}^{(n)} = \frac{(x_1^{(n+k-1)} - x_1^{(n+k-2)})^k x_k^{(n+1)} - (x_1^{(n+k)} - x_1^{(n+k-1)})^k x_k^{(n)}}{(x_1^{(n+k-1)} - x_1^{(n+k-2)})^k - (x_1^{(n+k)} - x_1^{(n+k-1)})^k} \quad (\text{II-16})$$

avec $x_1^{(n)} = s_n = n^{\text{ième}}$ terme de la suite à accélérer.

De plus on vérifie facilement que :

$$x_{k+1}^{(n)} = x_\infty + a_{k+1,k+1} d_{n-1}^{k+1} + \dots$$

avec :

$$a_{k+1,k+1} = \frac{a_1^k}{1 - a_1^k} \left(-k a_2 a_{kk} + (a_1 - 1) a_{k+1,k} \right)$$

on voit que pour $k > 1$ la transformation $\{x_k^{(n)}\} \rightarrow \{x_{k+1}^{(n)}\}$ est linéaire et on

a le résultat suivant :

Théorème II-5

Une condition suffisante pour que le procédé d'Aitken étendu (II-16) soit une transformation régulière est que la suite initiale $\{x_1^{(n)}\}$ soit convergente et que :

$$\lim_{n \rightarrow \infty} \frac{\Delta x_1^{(n)}}{\Delta x_1^{(n+1)}} \neq 1$$

Démonstration :

Si $\lim_{n \rightarrow \infty} \frac{\Delta x_1^{(n)}}{\Delta x_1^{(n+1)}} \neq 1$ les trois conditions du théorème de Toeplitz sont vérifiées. La transformation $\{x_k^{(n)}\} \rightarrow \{x_{k+1}^{(n)}\}$ est totale et de plus on a $\lim_{n \rightarrow \infty} x_2^{(n)} = x_\infty$ car

$$x_2^{(n)} = x_1^{(n)} + \frac{\Delta x_1^{(n-1)}}{1 - \frac{\Delta x_1^{(n-1)}}{\Delta x_1^{(n)}}} \quad \text{et } \lim_{n \rightarrow \infty} \Delta x_1^{(n-1)} = 0 \text{ puisque la}$$

suite initiale converge.

Ce procédé d'accélération de la convergence est particulièrement bien adapté aux suites obtenues par les méthodes itératives de résolution d'équation.

En effet soit à résoudre dans \mathbb{R} :

$$x = f(x)$$

Supposons f analytique autour de la racine a . On sait que :

$$f(a + \varepsilon) = a + a_1 \varepsilon + a_2 \varepsilon^2 + \dots$$

d'où

$$d_{n+1} = a_1 d_n + a_2 d_n^2 + \dots \text{ avec } a_n = \frac{f^{(n)}(a)}{n!}$$

Au paragraphe 6.1.3 nous donnerons un exemple qui montre la puissance de cette méthode dans ce cas .

Notons enfin que la quantité $x_{k+1}^{(n)} - x_k^{(n)}$ est une bonne approximation de la précision absolue sur $x_{k+1}^{(n)}$.

CHAPITRE III

ETUDE DE LA CONVERGENCE

Ce chapitre étudie la convergence des ε et ρ -algorithmes. Après avoir donné des théorèmes généraux de convergence de ces algorithmes on montre que l' ε -algorithme converge quand la suite initiale est soit totalement monotone, soit totalement oscillante, soit totalement oscillante décalée. On donne ensuite des conditions nécessaires et suffisantes d'accélération de la convergence, puis on introduit un paramètre d'accélération ce qui permet d'établir le lien entre l' ε -algorithme et le ρ -algorithme. On caractérise la valeur optimale de ce paramètre d'accélération. On compare ensuite ces méthodes au procédé de Richardson et enfin on explique les théorèmes d'accélération de la convergence et le paramètre d'accélération à l'aide des développements asymptotiques.

3-1 Théorèmes généraux de convergence

L'ε-algorithme et le ϑ^r-algorithme sont tous les deux de la forme :

$$\theta_{k+1}^{(n)} = \theta_{k-1}^{(n+1)} + D_k^{(n)} \quad D_k^{(n)} = \frac{a_k^{(n)}}{\theta_k^{(n+1)} - \theta_k^{(n)}} \quad (\text{II-1})$$

avec $a_k^{(n)} = 1$ pour l'ε-algorithme

et $a_k^{(n)} = x_{n+k+1}^r - x_n^r$ pour le ϑ^r-algorithme

Théorème III-1 Condition de convergence des algorithmes.

Supposons que $\lim_{n \rightarrow \infty} \theta_{2k}^{(n)} = S$. Une condition nécessaire et suffisante pour que

$$\lim_{n \rightarrow \infty} \theta_{2k+2}^{(n)} = S \text{ est que } \lim_{n \rightarrow \infty} D_{2k+1}^{(n)} = 0$$

La démonstration de ce théorème est évidente :

Considérons la suite des sommes partielles de la série :

$$\frac{1}{2} + \frac{1}{3} - \frac{5}{6} + \frac{1}{4} + \frac{1}{5} - \frac{9}{20} + \dots + \frac{1}{2n} + \frac{1}{2n+1} - \frac{4n+1}{2n(2n+1)} + \dots$$

La somme de cette série vaut zéro, mais, comme l'a montré Marx (39), $\epsilon_2^{(n)}$ ne converge pas mais oscille entre les valeurs 0 et 1. Il est facile de voir que la condition de convergence du théorème III-1 n'est pas vérifiée. On a :

$$D_1^{(n-1)} = \frac{\Delta S_n \cdot \Delta S_{n-1}}{\Delta S_{n-1} - \Delta S_n}$$

où S_n est la n^{ième} somme partielle de la série. Pour $n-1 = 3p-2$ et $n-1 = 3p-1$

on trouve que $\lim_{p \rightarrow \infty} D_1^{(n-1)} = 0$. Par contre pour $n-1 = 3p$ on a $D_1^{(3p)} = 1$.

Dans le même article Marx a donné une condition suffisante de convergence de $\epsilon_2^{(n)}$ pour n tendant vers l'infini. Nous avons repris et généralisé

cette condition :

Théorème III-2 Condition suffisante de convergence

Une condition suffisante pour que $\mathcal{E}_{2k+2}^{(n)}$ soit une transformation régulière est que :

$$\left. \begin{array}{l} \mathcal{E}_{2p}^{(n)} \geq \mathcal{E}_{2p}^{(n+1)} \geq 0 \\ \Delta \mathcal{E}_{2p}^{(n)} \leq \Delta \mathcal{E}_{2p}^{(n+1)} \leq 0 \end{array} \right\} \begin{array}{l} p = 0, \dots, k \text{ et} \\ \forall n \geq N \end{array}$$

démonstration :

Soit $\{S_n\}$ une suite convergente telle que pour tout $n \geq N$ on ait :

$$S_n \geq S_{n+1} \geq 0 \tag{III-2}$$

$$\Delta S_n \leq \Delta S_{n+1} \leq 0$$

La série équivalente à $\{s_n\}$ est $S_n = \sum_{i=0}^n u_i$ avec $u_0 = S_0$ et $u_{k+1} = S_{k+1} - S_k$

quelque soit k . Pour cette série la condition (III-2) devient :

$$u_n \leq u_{n+1} \leq 0 \text{ pour } n \geq N \geq 1 \tag{III-3}$$

l' \mathcal{E} -algorithme nous donne :

$$\mathcal{E}_2^{(n)} = S_{n+1} + \frac{1}{\frac{1}{u_{n+1}} - \frac{1}{u_n}}$$

Pour que $\mathcal{E}_2^{(n)}$ converge il faut et il suffit, d'après le théorème III-1, que

$$\lim_{n \rightarrow \infty} \left| \frac{1}{u_{n+1}} - \frac{1}{u_n} \right| = \infty$$

Montrons que la condition (III-3) entraîne que $\lim_{n \rightarrow \infty} \left(\frac{1}{u_{n+1}} - \frac{1}{u_n} \right) = -\infty$

Posons $v_n = 1/u_n$ et supposons que $A < v_{n+1} - v_n \leq 0$. on a :

$$A + v_n < v_{n+1} \leq 0$$

ou $A + v_n < \frac{1}{u_{n+1}} \leq 0$

$$u_{n+1} < \frac{1}{A+v_n} \leq 0$$

et de même

$$A + v_{n+1} < v_{n+2} \leq 0$$

$$2A+v_n < A+v_{n+1}$$

d'où

$$u_{n+2} < \frac{1}{2A+v_n} \leq 0$$

donc

$$\sum_{k=1}^{\infty} u_{n+k} < \sum_{k=1}^{\infty} \frac{1}{kA+v_n} \leq 0$$

puisque la série $\sum_{k=1}^{\infty} \frac{1}{kA+v_n}$ diverge il en est de même de $\sum_{k=1}^{\infty} u_{n+k}$ ce

qui est contraire aux hypothèses et par conséquent $A = -\infty$.

En général nous avons :

$$\begin{aligned} \varepsilon_{2k+2}^{(n)} &= \varepsilon_{2k}^{(n+1)} + \frac{1}{\varepsilon_{2k+1}^{(n+1)} - \varepsilon_{2k+1}^{(n)}} \\ &= \varepsilon_{2k}^{(n+1)} + \frac{1}{\varepsilon_{2k-1}^{(n+2)} - \varepsilon_{2k-1}^{(n+1)} + \frac{1}{\varepsilon_{2k}^{(n+2)} - \varepsilon_{2k}^{(n+1)}} - \frac{1}{\varepsilon_{2k}^{(n+1)} - \varepsilon_{2k}^{(n)}}} \end{aligned}$$

Supposons que $\varepsilon_{2p}^{(n)} \geq \varepsilon_{2p}^{(n+1)} \geq 0$ et que $\Delta \varepsilon_{2p}^{(n)} \leq \Delta \varepsilon_{2p}^{(n+1)} \leq 0$

pour $p = 0, \dots, k$ et $\forall n \geq N$. Alors :

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

et par conséquent $\lim_{n \rightarrow \infty} (\varepsilon_{2k+1}^{(n+1)} - \varepsilon_{2k+1}^{(n)}) = -\infty$ puisque

$$\lim_{n \rightarrow \infty} (\varepsilon_{2k-1}^{(n+2)} - \varepsilon_{2k-1}^{(n+1)}) = -\infty, \text{ ce qui démontre le théorème.}$$

3-2 Théorèmes particuliers de convergence

Nous allons maintenant démontrer la convergence de l' \mathcal{E} -algorithme dans des cas particuliers importants. Le résultat final auquel nous allons aboutir est le suivant (10) :

Théorème III-3 Si $\{S_n\}$ ou $\{-S_n\}$ est une suite soit totalement monotone, soit totalement oscillante, soit totalement oscillante décalée, si $\lim_{n \rightarrow \infty} S_n = S$ et si on applique l' \mathcal{E} -algorithme avec $\mathcal{E}_0^{(n)} = S_n$ alors $\lim_{n \rightarrow \infty} \mathcal{E}_{2^k}^{(n)} = S$ quelque soit k .

La suite de ce paragraphe est consacré à la démonstration du théorème III-3 ainsi qu'à l'étude de ses conséquences. Donnons tout d'abord quelques définitions et rappels (67) :

Définition III.1

On dit que la suite $\{S_n\}$ est totalement monotone si $(-1)^k \Delta^k S_n \geq 0$ pour $k = 0, 1, \dots$ et quelque soit n .

Définition III.2

On dit que la suite $\{S_n\}$ est totalement oscillante si la suite $\{(-1)^n S_n\}$ est totalement monotone

Définition III.3

On dit que la suite $\{S_n\}$ est totalement oscillante décalée s'il existe une constante finie non nulle S telle que la suite $\{S_n - S\}$ soit totalement oscillante.

Lemme III-1

Si la suite $\{S_n\}$ est totalement monotone alors la suite $\{(-1)^k \Delta^k S_n\}$ est totalement monotone ($k = 1, 2, \dots$)

Lemme III-2

Si la suite $\{S_n\}$ est totalement oscillante alors la suite $\{(-1)^k \Delta^k S_n\}$ est totalement oscillante ($k = 1, 2, \dots$)

Lemme III-3

Si la suite $\{S_n\}$ est totalement oscillante décalée alors la suite $\{(-1)^k \Delta^k S_n\}$ est totalement oscillante ($k=1, 2, \dots$)

Les démonstrations de ces trois lemmes sont évidentes. Elles reposent sur la définition même de l'opérateur Δ .

Lemme III-4

Si la suite $\{S_n\}$ est totalement monotone alors $H_k^{(n)}(S_n) \geq 0$

L'idée de la démonstration de ce lemme est la suivante : à partir de la théorie des moments de Hausdorff on montre que la suite $\{S_n\}$ est totalement monotone si et seulement si $S_n = \int_0^1 x^n dg(x)$ pour $n = 0, 1, \dots$ où $g(x)$ est une fonction bornée non décroissante dans $[0, 1]$. A partir de la théorie des moments de Stieltjes on montre que $H_k^{(0)}(S_0) \geq 0$ et $H_k^{(1)}(S_1) \geq 0$ si et seulement si $S_n = \int_0^\infty x^n d\bar{g}(x)$ pour $n = 0, 1, \dots$ où $\bar{g}(x)$ est une fonction bornée non décroissante sur $[0, +\infty[$. Dans ce dernier cas on a :

$$S_{n+p} = \int_0^\infty x^n d\bar{g}^{(p)}(x) \quad n, p = 0, 1, \dots$$

avec $d\bar{g}^{(p)}(x) = x^p d\bar{g}(x)$. $\bar{g}^{(p)}(x)$ est également bornée et non décroissante sur $[0, +\infty[$. Ainsi on a $H_k^{(n)}(S_n) \geq 0$ $k, n = 0, 1, \dots$. On obtient ensuite simplement le résultat désiré en posant $\bar{g}(x) = g(x)$ pour $x \in [0, 1]$ et $\bar{g}(x) = g(1)$ pour $x \in [1, +\infty[$.

A partir des lemmes III-1 et III-4 on voit que $H_k^{(n)}(\Delta^{2p} S_n) \geq 0$ et que $(-1)^k H_k^{(n)}(\Delta^{2p+1} S_n) \geq 0$ car k colonnes du déterminant de Hankel sont multipliées par -1 . D'où le résultat suivant :

Lemme III-5

Si on applique l' \mathcal{E} -algorithme à une suite $\{S_n\}$ totalement monotone alors :

$$\mathcal{E}_{2k}^{(n)} \geq 0 \quad \text{et} \quad \mathcal{E}_{2k+1}^{(n)} \leq 0 \quad n, k = 0, 1, \dots$$

La démonstration de ce lemme est évidente en sachant que $\mathcal{E}_{2k}^{(n)} = \frac{H_{k+1}^{(n)}(S_n)}{H_k^{(n)}(\Delta^2 S_n)}$

et que $\mathcal{E}_{2k+1}^{(n)} = \frac{H_k^{(n)}(\Delta^3 S_n)}{H_{k+1}^{(n)}(\Delta S_n)}$

Lemme III-6

Si la suite $\{S_n\}$ est totalement oscillante alors $(-1)^n H_k^{(n)}(S_n) \geq 0$

A partir des lemmes III-2 et III-6 on trouve que $(-1)^{kn} H_k^{(n)}(\Delta^{2p} S_n) \geq 0$

et que $(-1)^{k(n+1)} H_k^{(n)}(\Delta^{2p+1} S_n) \geq 0$. D'où

Lemme III-7

Si on applique l' \mathcal{E} -algorithme à une suite $\{S_n\}$ totalement oscillante alors :

$$(-1)^n \mathcal{E}_{2k}^{(n)} \geq 0 \quad \text{et} \quad (-1)^n \cdot \mathcal{E}_{2k+1}^{(n)} \leq 0 \quad n, k = 0, 1, \dots$$

On peut formuler des corollaires pour les lemmes III-5 et III-7

Corollaire III-1 :

Si la suite $\{-S_n\}$ est totalement monotone et si on applique l' \mathcal{E} -algorithme à

la suite $\{S_n\}$ alors $\mathcal{E}_{2k}^{(n)} \leq 0$ et $\mathcal{E}_{2k+1}^{(n)} \geq 0$ $n, k = 0, 1, \dots$

Corollaire III-2 :

Si la suite $\{-S_n\}$ est totalement oscillante et si on applique l' \mathcal{E} -algorithme

à la suite $\{S_n\}$ alors $(-1)^n \mathcal{E}_{2k}^{(n)} \leq 0$ et $(-1)^{n+1} \cdot \mathcal{E}_{2k+1}^{(n)} \geq 0$ $n, k = 0, 1, \dots$

La démonstration de ces corollaires est immédiate en utilisant le fait que si

$$\bar{\mathcal{E}}_0^{(n)} = a \mathcal{E}_0^{(n)} \quad \text{alors} \quad \bar{\mathcal{E}}_{2k}^{(n)} = a \mathcal{E}_{2k}^{(n)} \quad \text{et} \quad \bar{\mathcal{E}}_{2k+1}^{(n)} = \mathcal{E}_{2k+1}^{(n)} / a.$$

Pour les suites totalement oscillantes décalées le lemme III-7 et le corollaire

III-2 deviennent :

Lemme III-8

Si on applique l' ε -algorithme à une suite totalement oscillante décalée

$$\text{alors } (-1)^n (\varepsilon_{2k}^{(n)} - S) \geq 0 \quad \text{et} \quad (-1)^n \varepsilon_{2k+1}^{(n)} \leq 0 \quad n, k = 0, 1, \dots$$

Corollaire III-3 :

Si la suite $\{-S_n\}$ est totalement oscillante décalée et si on applique

l' ε -algorithme à la suite $\{S_n\}$ alors $(-1)^n (\varepsilon_{2k}^{(n)} - S) \leq 0$ et $(-1)^n \varepsilon_{2k+1}^{(n)} \geq 0$
 $n, k = 0, 1, \dots$

Les démonstrations de ce lemme et de son corollaire tiennent au fait que si

$$\varepsilon_0^{(n)} = \varepsilon_0^{(n)} + S \quad \text{alors} \quad \varepsilon_{2k}^{(n)} = \varepsilon_{2k}^{(n)} + S \quad \text{et} \quad \varepsilon_{2k+1}^{(n)} = \varepsilon_{2k+1}^{(n)}$$

Nous allons maintenant établir un résultat qui avait été énoncé à la fin du paragraphe 2.3.2 et qui est fondamental pour la démonstration du théorème

III-3 :

Lemme III-9

Les colonnes paires du tableau ε sont liées par la relation :

$$\varepsilon_{2k+2}^{(n)} - \varepsilon_{2k}^{(n)} = - \frac{[H_{k+1}^{(n)} (\Delta S_n)]^2}{H_{k+1}^{(n)} (\Delta^2 S_n) \cdot H_k^{(n)} (\Delta^2 S_n)} \quad n, k=0, 1, \dots$$

Démonstration :

Pour des déterminants d'ordre 4, le développement de Schweins du quotient de deux déterminants s'écrit (1) :

$$\begin{vmatrix} a_1 & b_1 & c_1 & d_1 \\ a_2 & b_2 & c_2 & d_2 \\ a_3 & b_3 & c_3 & d_3 \\ a_4 & b_4 & c_4 & d_4 \end{vmatrix} \cdot \begin{vmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{vmatrix} \cdot \begin{vmatrix} a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \\ a_4 & b_4 & c_4 \end{vmatrix} \cdot \begin{vmatrix} b_1 & c_1 & d_1 \\ b_2 & c_2 & d_2 \\ b_3 & c_3 & d_3 \end{vmatrix} \\ - \begin{vmatrix} b_2 & c_2 & d_2 \\ b_3 & c_3 & d_3 \\ b_4 & c_4 & d_4 \end{vmatrix} \cdot \begin{vmatrix} b_2 & c_2 \\ b_3 & c_3 \end{vmatrix} \cdot \begin{vmatrix} b_2 & c_2 & d_2 \\ b_3 & c_3 & d_3 \\ b_4 & c_4 & d_4 \end{vmatrix} \cdot \begin{vmatrix} b_2 & c_2 \\ b_3 & c_3 \end{vmatrix}$$

Si nous appliquons un développement identique à :

$$\varepsilon_{2k+2}^{(n)} - \varepsilon_{2k}^{(n)} = \frac{H_{k+2}^{(n)}(S_n)}{H_{k+1}^{(n)}(\Delta^2 S_n)} - \frac{H_{k+1}^{(n)}(S_n)}{H_k^{(n)}(\Delta^2 S_n)}$$

$$= \frac{\begin{vmatrix} S_n & \Delta S_n & \dots & \Delta^{k+1} S_n \\ \Delta S_n & \Delta^2 S_n & \dots & \Delta^{k+2} S_n \\ \dots & \dots & \dots & \dots \\ \Delta^{k+1} S_n & \Delta^{k+2} S_n & \dots & \Delta^{2k+2} S_n \end{vmatrix}}{\begin{vmatrix} S_n & \Delta S_n & \dots & \Delta^k S_n \\ \Delta S_n & \Delta^2 S_n & \dots & \Delta^{k+1} S_n \\ \dots & \dots & \dots & \dots \\ \Delta^k S_n & \Delta^{k+1} S_n & \dots & \Delta^{2k} S_n \end{vmatrix}} - \frac{\begin{vmatrix} \Delta^2 S_n & \Delta^3 S_n & \dots & \Delta^{k+2} S_n \\ \Delta^3 S_n & \Delta^4 S_n & \dots & \Delta^{k+3} S_n \\ \dots & \dots & \dots & \dots \\ \Delta^{k+2} S_n & \Delta^{k+3} S_n & \dots & \Delta^{2k+2} S_n \end{vmatrix}}{\begin{vmatrix} \Delta^2 S_n & \Delta^4 S_n & \dots & \Delta^{k+1} S_n \\ \Delta^3 S_n & \Delta^3 S_n & \dots & \Delta^{k+2} S_n \\ \dots & \dots & \dots & \dots \\ \Delta^{k+1} S_n & \Delta^{k+2} S_n & \dots & \Delta^{2k} S_n \end{vmatrix}}$$

on obtient :

$$\varepsilon_{2k+2}^{(n)} - \varepsilon_{2k}^{(n)} = \frac{\begin{vmatrix} \Delta S_n & \Delta^2 S_n & \dots & \Delta^{k+1} S_n \\ \Delta^2 S_n & \Delta^3 S_n & \dots & \Delta^{k+2} S_n \\ \dots & \dots & \dots & \dots \\ \Delta^{k+1} S_n & \Delta^{k+2} S_n & \dots & \Delta^{2k+1} S_n \end{vmatrix}^2}{\frac{H_{k+1}^{(n)}(\Delta^2 S_n) \cdot H_k^{(n)}(\Delta^2 S_n)}{\left[H_{k+1}^{(n)}(\Delta S_n) \right]^2}}$$

$$= \frac{H_{k+1}^{(n)}(\Delta^2 S_n) \cdot H_k^{(n)}(\Delta^2 S_n)}{H_{k+1}^{(n)}(\Delta S_n) \cdot H_k^{(n)}(\Delta S_n)}$$

Nous allons maintenant pouvoir démontrer le théorème III-3 :

1°) Suites totalement monotones

Si la suite $\{S_n\}$ est totalement monotone on sait, d'après les lemmes III-1

et III-4 que $H_{k+1}^{(n)} (\Delta^2 S_n)$ et $H_k^{(n)} (\Delta^2 S_n)$ sont positifs. Par conséquent, à cause du lemme III-9 on a :

$$\varepsilon_{2k+2}^{(n)} - \varepsilon_{2k}^{(n)} \leq 0 \text{ d'où d'après le lemme III-5 } 0 \leq \varepsilon_{2k+2}^{(n)} \leq \varepsilon_{2k}^{(n)}.$$

Or on sait que si on applique l' ε -algorithme à $\bar{\varepsilon}_0^{(n)} = \varepsilon_0^{(n)} - S$ on obtient

$\bar{\varepsilon}_{2k}^{(n)} = \varepsilon_{2k}^{(n)} - S$. Or la suite $\{\bar{\varepsilon}_0^{(n)}\}$ est totalement monotone et par conséquent $0 \leq \bar{\varepsilon}_{2k+2}^{(n)} \leq \bar{\varepsilon}_{2k}^{(n)}$. Donc la convergence vers 0 de $\bar{\varepsilon}_{2k}^{(n)}$ entraîne la convergence vers 0 de $\bar{\varepsilon}_{2k+2}^{(n)}$ et par conséquent si $\lim_{n \rightarrow \infty} \varepsilon_{2k}^{(n)} = S$ alors

$\lim_{n \rightarrow \infty} \varepsilon_{2k+2}^{(n)} = S$. De plus l'inégalité $0 \leq \varepsilon_{2k+2}^{(n)} \leq \varepsilon_{2k}^{(n)}$ nous assure que la

convergence de $\varepsilon_{2k+2}^{(n)}$ vers S est au moins aussi bonne que celle de $\varepsilon_{2k}^{(n)}$.

La démonstration est identique dans le cas où c'est la suite $\{-S_n\}$ qui est totalement monotone. On aura dans ce cas $\varepsilon_{2k}^{(n)} \leq \varepsilon_{2k+2}^{(n)} \leq 0$.

2°) Suites totalement oscillantes.

Si n est pair $H_{k+1}^{(n)} (\Delta^2 S_n)$ et $H_k^{(n)} (\Delta^2 S_n)$ sont de mêmes signes lorsque la suite $\{S_n\}$ est totalement oscillante d'après les lemmes III-2 et III-6. D'où, à cause du lemme III-9 $\varepsilon_{2k+2}^{(2p)} - \varepsilon_{2k}^{(2p)} \leq 0$ si $n=2p$, et d'après le lemme III-7 $0 \leq \varepsilon_{2k+2}^{(2p)} \leq \varepsilon_{2k}^{(2p)}$.

Si $n = 2p+1$ les deux déterminants sont des signes contraires et un raisonnement analogue nous conduit à $\varepsilon_{2k}^{(2p+1)} \leq \varepsilon_{2k+2}^{(2p+1)} \leq 0$.

Dans les deux cas la convergence de $\bar{\varepsilon}_{2k}^{(n)}$ vers 0 entraîne celle de $\bar{\varepsilon}_{2k+2}^{(n)}$ vers 0.

3°) Suites totalement oscillantes décalées.

En effectuant le même raisonnement sur la suite $\{S_n - S\}$ on arrive à la même conclusion que pour les suites totalement oscillantes. Si c'est la suite $\{-S_n\}$ qui est totalement monotone, ou totalement oscillante, ou

totallement oscillante décalée on est conduit à des résultats identiques ce qui achève la démonstration du théorème III-3.

Voyons maintenant les conséquences de ce théorème :

Corollaire III-4

Si on applique l' ε -algorithme à une suite totallement monotone alors

$$\varepsilon_{2k+1}^{(n)} \leq \varepsilon_{2k-1}^{(n)} \leq 0 \quad \forall n, k$$

et

$$\lim_{n \rightarrow \infty} \varepsilon_{2k+1}^{(n)} = -\infty \quad \forall k$$

démonstration :

Nous avons vu que :

$$0 \leq \varepsilon_{2k+2}^{(n)} = \varepsilon_{2k}^{(n+1)} + \frac{1}{\Delta \varepsilon_{2k+1}^{(n)}} \leq \varepsilon_{2k}^{(n)}$$

d'où

$$\Delta \varepsilon_{2k}^{(n)} \leq -\frac{1}{\Delta \varepsilon_{2k+1}^{(n)}} \quad \text{ou encore} \quad -\Delta \varepsilon_{2k+1}^{(n)} \geq \frac{1}{\Delta \varepsilon_{2k}^{(n)}}$$

ce qui entraîne

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

et par conséquent $\varepsilon_{2k+1}^{(n+1)} \leq \varepsilon_{2k-1}^{(n+1)} \leq 0$

or $\varepsilon_1^{(n)} = \frac{1}{\Delta S_n}$ d'où $\lim_{n \rightarrow \infty} \varepsilon_1^{(n)} = -\infty$ ce qui démontre le corollaire.

Si c'est la suite $\{-S_n\}$ qui est totallement monotone les conclusions du corollaire III-4 sont inversées. Il n'est pas possible de formuler de corollaire analogue pour les suites totallement oscillantes.

Comme je l'ai déjà fait remarquer le théorème III-3 n'assure pas une convergence de $\varepsilon_{2k+2}^{(n)}$ plus rapide que celle de $\varepsilon_{2k}^{(n)}$. Pour s'en convaincre

il n'y a qu'à considérer la suite totalement monotone $\left\{ S_n = 1 + \frac{1}{n+1} \right\}$.

on trouve, après des calculs pénibles, que :

$$\varepsilon_{2k}^{(n)} = \frac{S_{n+k} + k}{k+1}$$

et que
$$\lim_{n \rightarrow \infty} \frac{\varepsilon_{2k+2}^{(n)} - 1}{\varepsilon_{2k}^{(n)} - 1} = \frac{k+1}{k+2} \quad \forall k$$

3-3 Accélération de la convergence

Soit $E = \mathbb{C}$ l'espace des suites de nombres réels ou complexes. Munissons \mathbb{C} de la norme $\|x\| = \sup_i |x_i|$ où x_i désigne le $i^{\text{ème}}$ terme de la suite $x \in \mathbb{C}$.

Prenons pour x'_n la forme linéaire qui, à une suite, associe la différence $x_{n+1} - x_n$. La suite $\{x'_n\}$ converge donc faiblement vers $x'_\infty \in (\mathbb{C})' = l_1$. Avec ces notations et en utilisant la définition 0-2 nous dirons que $x \in \mathbb{C}$ converge plus vite que $y \in \mathbb{C}$ par rapport à $\{x'_n\}$ si :

$$\lim_{n \rightarrow \infty} \frac{\Delta x_n}{\Delta y_n} = 0$$

Théorème III-4 : Condition d'accélération de la convergence

Soit un algorithme de la forme (III-4). Supposons que $\lim_{n \rightarrow \infty} \theta_{2k}^{(n)} = \lim_{n \rightarrow \infty} \theta_{2k+2}^{(n)} = S$

Une condition nécessaire et suffisante pour que $\theta_{2k+2}^{(n)}$ converge plus vite que

$\theta_{2k}^{(n+1)}$ est que :

$$\lim_{n \rightarrow \infty} \frac{\Delta D_{2k+1}^{(n)}}{\Delta \theta_{2k}^{(n+1)}} = -1$$

démonstration :

On a $\Delta \theta_{2k+2}^{(n)} = \Delta \theta_{2k}^{(n+1)} + \Delta D_{2k+1}^{(n)}$. Supposons que $\lim_{n \rightarrow \infty} \frac{\Delta \theta_{2k+2}^{(n)}}{\Delta \theta_{2k}^{(n+1)}} = 0$. Cela

entraîne que $\lim_{n \rightarrow \infty} \left(1 + \frac{\Delta D_{2k+1}^{(n)}}{\Delta \theta_{2k}^{(n+1)}} \right) = 0$. Inversement si la condition du

théorème est vérifiée $\theta_{2k+2}^{(n)}$ convergera plus vite que $\theta_{2k}^{(n+1)}$

On peut déduire de ce théorème un résultat intéressant. Nous avons :

$$e_{k+1}^{(n)} = e_k^{(n+1)} + D_{2k+1}^{(n)} \quad \text{avec} \quad e_k^{(n)} = \theta_{2k}^{(n)} - S.$$

Si $\theta_{2k+2}^{(n)}$ tend vers S alors $\lim_{n \rightarrow \infty} (e_k^{(n+1)} + D_{2k+1}^{(n)}) = 0$. Si $\theta_{2k+2}^{(n)}$ tend vers S plus vite que $\theta_{2k}^{(n+1)}$ alors on aura :

$$\lim_{n \rightarrow \infty} \frac{D_{2k+1}^{(n)}}{e_k^{(n+1)}} = -1 \text{ et par conséquent, pour } n \text{ suffisamment grand on aura}$$

$$e_k^{(n+1)} \sim -D_{2k+1}^{(n)}$$

ce qui fournit un ordre de grandeur de l'erreur.

Considérons maintenant une suite pour laquelle la condition du théorème III-4 n'est pas vérifiée pour une certaine valeur de k. Afin d'accélérer la convergence nous allons introduire un facteur d'accélération un peu dans l'idée du paramètre de sur-relaxation de la méthode de Gauss-Seidel pour la résolution des systèmes linéaires. Nous allons définir $\bar{\theta}_{2k+2}^{(n)}$ par :

$$\bar{\theta}_{2k+2}^{(n)} = \theta_{2k}^{(n+1)} + \omega_k D_{2k+1}^{(n)} \quad (\text{III-4})$$

Théorème III-5 : Choix du facteur d'accélération

Une condition nécessaire et suffisante pour que $\bar{\theta}_{2k+2}^{(n)}$ converge plus vite que $\theta_{2k+2}^{(n)}$ est de prendre $\omega_k = \lim_{n \rightarrow \infty} \frac{\Delta \theta_{2k}^{(n+1)}}{\Delta D_{2k+1}^{(n)}}$

Démonstration :

$$\begin{aligned} \text{Nous avons} \quad \bar{\theta}_{2k+2}^{(n)} &= \theta_{2k}^{(n+1)} + \omega_k D_{2k+1}^{(n)} \\ &= \theta_{2k}^{(n+1)} + D_{2k+1}^{(n)} + (\omega_k - 1) D_{2k+1}^{(n)} \\ &= \theta_{2k+2}^{(n)} + (\omega_k - 1) D_{2k+1}^{(n)} \end{aligned}$$

$\bar{\theta}_{2k+2}^{(n)}$ convergera plus vite que $\theta_{2k+2}^{(n)}$ si $\lim_{n \rightarrow \infty} \frac{\Delta \bar{\theta}_{2k+2}^{(n)}}{\Delta \theta_{2k+2}^{(n)}} = 0$. D'où

$$\lim_{n \rightarrow \infty} \left(1 + (\omega_k - 1) \frac{\Delta D_{2k+1}^{(n)}}{\Delta \theta_{2k+2}^{(n)}} \right) = 0$$

Ce qui peut encore s'écrire :

$$\lim_{n \rightarrow \infty} (\omega_k - 1) \frac{\Delta D_{2k+1}^{(n)}}{\Delta \theta_{2k+2}^{(n)}} = -1$$

ou

$$\lim_{n \rightarrow \infty} \frac{\Delta D_{2k+1}^{(n)}}{\Delta \theta_{2k+2}^{(n)}} = \frac{1}{1 - \omega_k}$$

or $\Delta \theta_{2k+2}^{(n)} = \Delta \theta_{2k}^{(n+1)} + \Delta D_{2k+1}^{(n)}$ d'où finalement $\omega_k = - \lim_{n \rightarrow \infty} \frac{\Delta \theta_{2k}^{(n+1)}}{\Delta D_{2k+1}^{(n)}}$

Inversement il est facile de voir que ce choix de ω_k entraîne une convergence de $\bar{\theta}_{2k+2}^{(n)}$ plus rapide que celle de $\theta_{2k+2}^{(n)}$.

On voit que si la condition du théorème III-4 est vérifiée alors on a $\omega_k = 1$ et il est impossible d'accélérer la convergence en utilisant un algorithme de la forme (III-4).

Ce paramètre ω_k apparait comme le lien entre ϵ et ξ -algorithmes. En effet, si en prenant $\left\{ S_n = 1 + \frac{1}{n+1} \right\}$ on trouve $\omega_0 = 2$ avec $a_k^{(n)} = 1$ ce qui redonne le ξ -algorithme. Inversement toujours avec $a_k^{(n)} = 1$ et avec

$\left\{ S_n = s + a b^n \right\}$ on obtient $\omega_0 = 1$ qui n'est autre que l' ϵ -algorithme. Il serait évidemment intéressant de trouver un algorithme de la forme (III-4) qui soit tel que $\theta_{2k}^{(n)} = s$ si $\theta_0^{(n)} = s + \sum_{i=1}^k a_i b_i^n$ et tel que $\theta_{2k}^{(n)} = a_k / b_k$

si $\theta_0^{(n)} = \frac{\sum_{i=0}^k a_i n^i}{\sum_{i=0}^k b_i n^i}$. On montre, après des calculs pénibles, qu'un tel algorithme ne peut exister car il y a incomptabilité entre les conditions obtenues en imposant à la fois ces deux propriétés. On arrive également à la même conclusion pour un algorithme de la forme (III-4) qui

utiliserait des relations différentes pour les colonnes paires et pour les colonnes impaires. Cependant en prenant $a_k^{(n)} = 1$ et $\omega_k = k+2$ on obtient les résultats suivants :

$$\text{si } \theta_0^{(n)} = \frac{an + b}{cn + d} \quad \rho_{2k}^{(n)} = \frac{a}{c} \quad \forall k > 0$$

$$\theta_{2k}^{(n)} = \frac{\theta_0^{(n+k)} + k(k+2) \frac{a}{c}}{(k+1)^2} \quad \varepsilon_{2k}^{(n)} = \frac{\theta_0^{(n+k)} + k \frac{a}{c}}{k+1}$$

$$\frac{\theta_{2k}^{(n)} - \frac{a}{c}}{\varepsilon_{2k}^{(n)} - \frac{a}{c}} = \frac{1}{k+1}$$

$$\text{si } \theta_0^{(n)} = s + ab^n \quad \varepsilon_{2k}^{(n)} = s \quad \forall k > 0$$

$$\theta_{2k}^{(n)} = s + (-1)^k \frac{a}{k+1} b^{n+k} \quad \rho_{2k}^{(n)} = s + (-1)^k ab^{n+k}$$

$$\frac{\theta_{2k}^{(n)} - s}{\rho_{2k}^{(n)} - s} = \frac{1}{k+1}$$

Ainsi, dans les deux cas, la convergence est améliorée et le facteur d'amélioration est le même.

3-4 Comparaison avec le procédé de Richardson

Au lieu d'étudier S_n quand n tend vers l'infini, considérons, comme le fait Laurent (36), l'extrapolation de la fonction $S(h)$ en $h = 0$:

$$S_n = S\left(\frac{A}{n}\right) = S(h)$$

Supposons que $S(h)$ possède k dérivées sur $[0, H]$ (dérivées à droite en zéro)

on a :

$$S(h) = S(o) + h S'(o) + \frac{h^2}{2!} S''(o) + \dots + \frac{h^{k-1}}{(k-1)!} S^{(k-1)}(o) + h^k v(h)$$

où V est une fonction bornée de h dans $(0, H)$.

Soit $\{h_n\}$ une suite strictement décroissante et tendant vers 0 quand n tend vers l'infini. Prenons $S_n \equiv S(h_n)$ et cherchons à quelles conditions les termes en h disparaissent des développements de $\mathcal{E}_2^{(n)}$ et $\mathcal{P}_2^{(n)}$. Pour l' \mathcal{E} -algorithme on trouve que $h_{n+1}^2 = h_n \cdot h_{n+2}$ c'est-à-dire que l'on a avantage à l'appliquer à une suite $\{S_{n_i}\}$ où les n_i sont en progression géométrique. On aura alors

$$\mathcal{E}_2^{(n)} = S(o) + \frac{h_{n+1}^2}{2!} S''(o) + \dots$$

Pour la forme simplifiée (II-8) du \mathcal{P} -algorithme on trouve que $\frac{2}{h_{n+1}} = \frac{1}{h_n} + \frac{1}{h_{n+2}}$ c'est-à-dire que les n_i doivent être en progression arithmétique. On a alors $\mathcal{P}_2^{(n)} = S(o) + \frac{h_{n+1}^2}{4} S''(o) + \dots$

Laurent (36) a montré que si $S_n = S + \frac{B}{n^2} + \frac{D(n)}{n^4}$ où D est une fonction bornée de n , alors l'application du procédé de Richardson à S_{n_1} et S_{n_2} avec $n_2 = kn_1$ fournit une valeur S_1^* telle que :

$$S_1^* - S = \frac{1}{n_1^4} \cdot \frac{1}{1-k^2} \left(D(n_1) - \frac{D(n_2)}{k^2} \right)$$

Il est facile de voir que l'application du procédé de Richardson à S_1^* et S_2^* (où S_2^* est la valeur extrapolée sur S_{n_2} et S_{n_3} avec $n_3 = kn_2$) fournit une valeur S_1^{**} telle que :

$$S_1^{**} - S = \frac{1}{n_1^4} \cdot \frac{1}{(1-k^2)^2} \left(D(n_1) - 2 \frac{D(n_2)}{k^2} + \frac{D(n_3)}{k^4} \right) + O\left(\frac{1}{n_1^6}\right)$$

si on applique l' \mathcal{E} -algorithme à S_{n_1} , S_{n_2} et S_{n_3} on obtient une valeur $\mathcal{E}_2^{(o)}$ telle que :

$$\mathcal{E}_2^{(o)} - S = \frac{1}{n_1^4} \cdot \frac{1}{(1-k^2)^2} \left(D(n_1) - 2 \frac{D(n_2)}{k^2} + \frac{D(n_3)}{k^4} \right) + O\left(\frac{1}{n_1^6}\right)$$

Ce choix des n_i en progression arithmétique ou géométrique suivant l'algorithme utilisé est à rapprocher des transformations de condensation étudiées par Daniel (19) et par Wijngaarden (55). On peut également le considérer comme un choix optimal des n_i et rapprocher ces résultats de ceux de Baranger sur les formules optimales pour le calcul de la somme d'une série (4).

3-5 Liaison avec les développements asymptotiques.

Tous les résultats que nous venons d'établir s'expliquent facilement si l'on considère les $\omega_k D_{2k+1}^{(n)}$ comme les termes successifs d'un développement asymptotique (7, 20). Soit $\{S_n\}$ une suite qui converge vers S . Soit G' l'ensemble des suites $D_{2k+1}^{(n)}$ $k = 0, 1, \dots$. Supposons que $D_{2k+1}^{(n)} = o(D_{2k-1}^{(n+1)})$ ce qui nécessite, notons le au passage, que $D_{2k-1}^{(n)} = o(1) \forall k$ qui n'est autre que la condition du théorème III₁. Alors, s'il vérifie cette propriété, l'ensemble G' est la restriction à \mathbb{N} des fonctions définies sur $[x_0, +\infty[$ d'une échelle de comparaison G .

Supposons que $T_{n+k} = S - S_{n+k}$ possède un développement asymptotique jusqu'à l'ordre k au voisinage de $+\infty$ par rapport à G' et que ce développement puisse s'écrire :

$$T_{n+k} = \sum_{i=1}^k \omega_{i-1} D_{2i-1}^{(n+k-i)} + o(D_{2k-1}^{(n)})$$

le problème est de trouver les constantes ω_i . On a :

$$S - S_{n+k} = \sum_{i=1}^k \omega_{i-1} D_{2i-1}^{(n+k-i)} + o(D_{2k-1}^{(n)})$$

d'où

$$\begin{aligned} S &= \theta_{2k}^{(n)} + o(D_{2k-1}^{(n)}) \\ &= \theta_{2k-2}^{(n+1)} + \omega_{k-1} D_{2k-1}^{(n)} + o(D_{2k-1}^{(n)}) \end{aligned}$$

on choisit ω_{k-1} de façon à avoir :

$$0 = \Delta \theta_{2k-2}^{(n+1)} + \omega_{k-1} \Delta D_{2k-1}^{(n)} + o(\Delta D_{2k-1}^{(n)})$$

ce qui entraînera $S = \theta_{2k}^{(n)} + o(D_{2k-1}^{(n)})$ puisque $\sum_{i=n}^{\infty} \Delta \theta_{2k-2}^{(i)}$ et $\sum_{i=n}^{\infty} \Delta D_{2k-1}^{(i)}$ sont convergentes $\forall n$. On a donc :

$$\omega_{k-1} \cdot \Delta D_{2k-1}^{(n)} = - \Delta \theta_{2k-2}^{(n+1)} + o(\Delta D_{2k-1}^{(n)})$$

d'où

$$\omega_{k-1} = - \lim_{n \rightarrow \infty} \frac{\Delta \theta_{2k-2}^{(n+1)}}{\Delta D_{2k-1}^{(n)}}$$

ce qui n'est autre que la condition du théorème III-5. Ainsi le théorème III-4 apparaît comme une condition nécessaire et suffisante pour que $\omega_k = 1$. Le fait que le choix de ω_k donné par le théorème III-5 fournit le plus rapide de tous les algorithmes de la forme III-4 signifie simplement que ω_k est le coefficient de $D_{2k+1}^{(n)}$ dans le développement asymptotique de $S - S_{n+k}$. Ce choix de ω_k donne le seul algorithme de la classe pour lequel on ait :

$$S - \theta_{2k}^{(n)} = o(D_{2k-1}^{(n)})$$

De cette comparaison avec les développements asymptotiques on peut déduire deux conséquences pratiques :

1°) nous avons : $S - \theta_{2k-2}^{(n+1)} = \omega_{k-1} D_{2k-1}^{(n)} + o(D_{2k-1}^{(n)})$

donc $S - \theta_{2k-2}^{(n+1)} \sim \omega_{k-1} D_{2k-1}^{(n)}$

et par conséquent $S - \theta_{2k-2}^{(n+1)} = o(D_{2k-1}^{(n)})$

Donc si on arrête le développement asymptotique à l'indice $k - 1$ l'erreur sera de l'ordre de grandeur de $D_{2k-1}^{(n)}$. On retrouve ainsi le résultat obtenu au paragraphe 3.3

2°) $\theta_{2k+2}^{(n)}$ est une meilleure approximation de S que $\theta_{2k}^{(n+1)}$.

CHAPITRE IV

FORMES CONFLUENTES DE L' ε -ALGORITHME

Soit une fonction d'une variable. Le problème que l'on cherche à résoudre dans ce chapitre est celui de l'estimation de la valeur de la fonction pour une valeur infinie de la variable à partir de la valeur de la fonction et de ses dérivées en un point. On utilise pour cela les formes confluentes des ε et ξ -algorithmes. Après avoir posé ce problème de façon générale on étudie les première et seconde formes confluentes de l' ε -algorithme. On donne des théorèmes de convergence, d'accélération de la convergence et de caractérisation des fonctions pour lesquelles on peut obtenir un résultat exact. On introduit successivement dans ces formes confluentes un paramètre puis une fonction d'accélération ce qui permet d'effectuer la liaison avec le ξ -algorithme. On explique les résultats obtenus à l'aide des développements asymptotiques.

Soit E un espace vectoriel topologique et E' son dual topologique. Soit $x'(t)$ un élément de E' qui dépend continuellement du paramètre t pour $t \in [T, +\infty[$ et qui converge faiblement vers $x'(\infty)$:

$$\lim_{t \rightarrow \infty} \langle x, x'(t) \rangle = \langle x, x'(\infty) \rangle \quad \forall x \in E$$

Le problème que nous ^{nous} posons maintenant est celui de l'estimation de cette limite à partir de $\langle x, x'(t) \rangle$ que l'on connaît pour une certaine valeur finie de t . Ce problème, on le voit, est une généralisation de celui abordé au chapitre II en ce sens que le premier apparaît comme la restriction à \mathbb{N} du second. On peut le formuler ainsi :

Soit $\varphi(t, a)$ une fonction donnée, continuellement dérivable par rapport à t autant de fois qu'il sera nécessaire :

$$\varphi : [T, +\infty[\times P \longrightarrow \mathbb{R} \quad P \subseteq \mathbb{R}^k \quad \text{telle que } \forall a \in P$$

$$\lim_{t \rightarrow \infty} \varphi(t, a) = 0$$

soit $V_\varphi \{x'(t)\}$ la variété de E des x tels que :

$$\langle x, x'(\infty) - x'(t) \rangle = \varphi(t, a) \quad \forall t \in [T, +\infty[$$

Supposons $\varphi(t, a)$ avoir la propriété suivante :

$$\forall \alpha = (\alpha_1, \dots, \alpha_k) \in \mathbb{R}^k \text{ donné et assez voisin de } 0 \in \mathbb{R}^k \text{ alors } \exists t_0$$

tel que $\forall t \geq t_0$ le système :

$$\left. \begin{array}{l} \alpha_1 = \varphi'(t, a) \\ \alpha_2 = \varphi''(t, a) \\ \dots \\ \alpha_k = \varphi^{(k)}(t, a) \end{array} \right\} \quad (\text{IV-1})$$

admette une et une seule solution a .

solution de (IV-1) est encore assurée par le théorème II-1.

Nous allons maintenant étudier deux tels procédés : ce sont les formes confluentes de l' \mathcal{E} -algorithme qui supposent que la fonction $\varphi(t, a)$ est une somme d'exponentielles en t analogue à la forme du théorème II-2.

4-1 Première forme confluyente de l' \mathcal{E} -algorithme

Soit la forme habituelle (II-5) de l' \mathcal{E} -algorithme. On remplace la variable discrète n par la variable continue $t = a + n \cdot \Delta t$, $\mathcal{E}_{2k+1}^{(n)}$ par $\mathcal{E}_{2k+1}(t) / \Delta t$ et $\mathcal{E}_{2k}^{(n)}$ par $\mathcal{E}_{2k}(t)$, puis on fait tendre Δt vers zéro. On obtient alors la première forme confluyente de l' \mathcal{E} -algorithme (62) :

$$\mathcal{E}_{k+1}(t) = \mathcal{E}_{k-1}(t) + \frac{1}{\mathcal{E}_k(t)} \quad k=0,1,\dots \quad (IV-4)$$

avec $\mathcal{E}_{-1}(t) = 0$ et $\mathcal{E}_0(t) = \langle x, x'(t) \rangle$

Les propriétés de (IV-4) sont identiques à celles de (II-5); c'est-à-dire

que si $\bar{\mathcal{E}}_0(t) = a \mathcal{E}_0(t) + b$ alors $\bar{\mathcal{E}}_{2k}(t) = \mathcal{E}_{2k}(t)a + b$ et que $\bar{\mathcal{E}}_{2k+1}(t) = \mathcal{E}_{2k+1}(t) / a$.

Soit $H_k^{(n)}$ le déterminant fonctionnel de Hankel suivant :

$$\begin{array}{lll} D^n \langle x, x'(t) \rangle & D^{n+1} \langle x, x'(t) \rangle & \dots \dots \dots D^{n+k-1} \langle x, x'(t) \rangle \\ D^{n+1} \langle x, x'(t) \rangle & D^{n+2} \langle x, x'(t) \rangle & \dots \dots \dots D^{n+k} \langle x, x'(t) \rangle \\ \hline D^{n+k-1} \langle x, x'(t) \rangle & D^{n+k} \langle x, x'(t) \rangle & \dots \dots \dots D^{n+k-2} \langle x, x'(t) \rangle \end{array}$$

avec $H_0^{(n)} = 1 \quad \forall n$. On montre que (62) :

$$\varepsilon_{2k}(t) = \frac{H_{k+1}^{(0)}}{H_k^{(2)}} \qquad \varepsilon_{2k+1}(t) = \frac{H_k^{(3)}}{H_{k+1}^{(1)}} \qquad (IV-5)$$

$$\varepsilon_{2k+2}(t) = \varepsilon_{2k}(t) - \frac{\left[\frac{H_{k+1}^{(1)}}{H_{k+1}^{(2)}} \right]^2}{H_k^{(2)} H_{k+1}^{(2)}}$$

Wynn (57) a montré que sous certaines conditions on avait $\varepsilon_{2k}(t) = \langle x, x^{\infty} \rangle$
 Ce sont ces conditions que nous allons maintenant étudier.

4.0.1 Exactitude de $\varepsilon_{2k}(t)$

D'après la définition 0-5 on dira que $\varepsilon_{2k}(t)$ est exacte pour
 $t \geq t_0$ si $\varepsilon_{2k}(t) = \langle x, x^{\infty} \rangle \quad \forall t \geq t_0$.

Afin de simplifier l'écriture de ce qui suit nous poserons

$$\langle x, x^{\infty}(t) \rangle = f(t) = f$$

$$\text{et } \langle x, x^{\infty} \rangle = S$$

Lemme IV-1

Une condition nécessaire et suffisante pour que $\varepsilon_{2k}(t) = S$ est que :

$$\begin{vmatrix} f - S & f' & \dots & f^{(k)} \\ f'' & f''' & \dots & f^{(k+1)} \\ \dots & \dots & \dots & \dots \\ f^{(k)} & f^{(k+1)} & \dots & f^{(2k)} \end{vmatrix} = 0$$

démonstration : On a d'après (IV-5)

$$\mathcal{E}_{2k}(t) = S = \frac{\begin{vmatrix} f & \dots & f^{(k)} \\ \dots & \dots & \dots \\ f^{(k)} & \dots & f^{(2k)} \end{vmatrix}}{\begin{vmatrix} f'' & \dots & f^{(k+1)} \\ \dots & \dots & \dots \\ f^{(k+1)} & \dots & f^{(2k)} \end{vmatrix}} = \frac{\begin{vmatrix} f & \dots & f^{(k)} \\ \dots & \dots & \dots \\ f^{(k)} & \dots & f^{(2k)} \end{vmatrix}}{\begin{vmatrix} 1 & f' & \dots & f^{(k)} \\ \dots & \dots & \dots & \dots \\ 0 & f'' & \dots & f^{(k+1)} \\ \dots & \dots & \dots & \dots \\ 0 & f^{(k+1)} & \dots & f^{(2k)} \end{vmatrix}}$$

d'où

$$\begin{vmatrix} f - S & f' & \dots & f^{(k)} \\ f' & f'' & \dots & f^{(k+1)} \\ \dots & \dots & \dots & \dots \\ f^{(k)} & f^{(k+1)} & \dots & f^{(2k)} \end{vmatrix} = 0 \text{ et inversement}$$

Lemme IV-2

Une condition nécessaire et suffisante pour que $\mathcal{E}_{2k}(t)$ soit exacte pour $t \geq t_0$ est que $g(t) = f(t) - S$ vérifie une équation différentielle linéaire d'ordre k à coefficients constants pour $t \geq t_0$.

démonstration :

$$\begin{vmatrix} g & g' & \dots & g^{(k)} \\ g' & g'' & \dots & g^{(k+1)} \\ \dots & \dots & \dots & \dots \\ g^{(k)} & g^{(k+1)} & \dots & g^{(2k)} \end{vmatrix} = 0$$

si et seulement s'il existe $\alpha_0, \alpha_1, \dots, \alpha_k$ tels que :

$$\alpha_0 g^{(k)} + \alpha_1 g^{(k-1)} + \dots + \alpha_k g = 0$$

$$\alpha_0 g^{(k+1)} + \alpha_1 g^{(k)} + \dots + \alpha_k g' = 0$$

.....

$$\alpha_0 g^{(2k)} + \alpha_1 g^{(2k-1)} + \dots + \alpha_k g^{(k)} = 0$$

donc $g, g', \dots, g^{(k)}$ sont solutions de :

$$\alpha_0 y^{(k)} + \alpha_1 y^{(k-1)} + \dots + \alpha_k y = 0$$

Les solutions de cette équation différentielle forment un espace vectoriel de dimension inférieure ou égale à k . Par conséquent $g, g', \dots, g^{(k)}$ sont linéairement dépendants et il existe des constantes c_1, \dots, c_k telles que :

$$g^{(k)} + c_1 g^{(k-1)} + \dots + c_k g = 0$$

Théorème IV-1

Une condition nécessaire et suffisante pour que $\mathcal{E}_{2k}(t)$ soit exacte pour $t \geq t_0$ est que $f(t) = S + \sum_{i=1}^p A_i(t) e^{s_i t} + \sum_{i=p+1}^q [B_i(t) \cos b_i t +$

$C_i(t) \sin b_i t] e^{s_i t}$ où A_i, B_i et C_i sont des polynômes en t tels que si d_i est égal au degré de A_i plus un pour $i = 1, \dots, p$ et au plus grand des degrés de B_i et de C_i plus un pour $i = p+1, \dots, q$, on ait :

$$\sum_{i=1}^p d_i + 2 \sum_{i=p+1}^q d_i = k$$

La démonstration de ce théorème est évidente car $f(t)$ est la solution générale d'une équation différentielle linéaire d'ordre k à coefficients

constants d'après le lemme IV-2. On voit que ce théorème est l'analogie du théorème II-2 correspondant à la forme discrète de l' ε -algorithme.

4.1.2 Convergence de $\varepsilon_{2k}(t)$

Nous allons maintenant donner des conditions nécessaires et suffisantes de convergence de $\varepsilon_{2k}(t)$ vers S et d'accélération de la convergence au sens de la définition 0-2 : On dira que $\varepsilon_{2k+2}(t)$ converge plus vite que $\varepsilon_{2k}(t)$ si $\lim_{t \rightarrow \infty} \frac{\varepsilon'_{2k+2}(t)}{\varepsilon'_{2k}(t)} = 0$

Théorème IV-2

Supposons que $\lim_{t \rightarrow \infty} \varepsilon_{2k-2}(t) = S$. Une condition nécessaire et suffisante

pour que $\lim_{t \rightarrow \infty} \varepsilon_{2k}(t) = S$ est que :

$$\lim_{t \rightarrow \infty} \frac{\left[\begin{matrix} (1) \\ H_k \end{matrix} \right]^2}{H_{k-1}^{(2)} \cdot H_k^{(2)}} = 0$$

La démonstration de ce théorème est immédiate. Elle découle de la troisième des relations (IV-5). On a également le résultat suivant :

Théorème IV-3

Une condition nécessaire et suffisante pour que $\lim_{t \rightarrow \infty} \varepsilon_{2k}(t) = S$ est que :

$$\lim_{t \rightarrow \infty} \frac{H_{k+1}^{(0)} - f \cdot H_k^{(2)}}{H_k^{(2)}} = 0$$

La démonstration découle de la première des relations (IV-5). Les conditions des théorèmes IV-2 et IV-3 sont naturellement équivalentes. On utilisera celle qui nécessitera le moins de calculs en remarquant que le théorème IV-2 suppose vérifiée la convergence de $\varepsilon_{2k-2}(t)$ vers S.

Nous allons maintenant étudier des conditions d'accélération de la convergence. Auparavant nous devons établir le résultat suivant :

Lemme IV-3

On a les relations suivantes :

$$H_{k+2}^{(n-1)} \circ H_k^{(n+1)} + \left[H_{k+1}^{(n)} \right]^2 = H_{k+1}^{(n-1)} \circ H_{k+1}^{(n+1)} \quad (\text{IV-6})$$

$$\varepsilon'_{2k+1}(t) = - \frac{H_k^{(2)} \circ H_{k+1}^{(2)}}{\left[H_{k+1}^{(1)} \right]^2} \quad (\text{IV-7})$$

$$\varepsilon'_{2k}(t) = \frac{H_k^{(1)} \circ H_{k+1}^{(1)}}{\left[H_k^{(2)} \right]^2} \quad (\text{IV-8})$$

démonstration : d'après les relations (IV-5) on a :

$$\frac{H_{k+2}^{(0)}}{H_{k+1}^{(2)}} - \frac{H_{k+1}^{(0)}}{H_k^{(2)}} = - \frac{\left[H_{k+1}^{(1)} \right]^2}{H_k^{(2)} \circ H_{k+1}^{(2)}}$$

$$\text{d'où } H_{k+2}^{(0)} \circ H_k^{(2)} + \left[H_{k+1}^{(1)} \right]^2 = H_{k+1}^{(0)} \circ H_{k+1}^{(2)}$$

on obtient immédiatement (IV-6) en remplaçant f par $f^{(n-1)}$

La troisième des relations (IV-5) est équivalente à :

$$\varepsilon_{2k+2}(t) = \varepsilon_{2k}(t) + \frac{1}{\varepsilon'_{2k+1}(t)}$$

d'où la relation (IV-7). Si maintenant dans (IV-6) on fait $n=2$, on obtient :

$$H_{k+2}^{(1)} \circ H_k^{(3)} - H_{k+1}^{(1)} \circ H_{k+1}^{(3)} = - \left[H_{k+1}^{(2)} \right]^2$$

Divisons les deux membres par $H_{k+1}^{(1)} \cdot H_{k+2}^{(1)}$

$$\frac{H_k^{(3)}}{H_{k+1}^{(1)}} - \frac{H_{k+1}^{(3)}}{H_{k+2}^{(1)}} = - \frac{[H_{k+1}^{(2)}]^2}{H_{k+1}^{(1)} \cdot H_{k+2}^{(1)}}$$

d'où en utilisant (IV-5) :

$$\varepsilon_{2k+1}(t) - \varepsilon_{2k+3}(t) = - \frac{[H_{k+1}^{(2)}]^2}{H_{k+1}^{(1)} \cdot H_{k+2}^{(1)}} = - \frac{1}{\varepsilon'_{2k+2}(t)}$$

Théorème IV-4

Une condition nécessaire et suffisante pour que $\varepsilon_{2k}(t)$ converge vers S plus vite que $\varepsilon_{2k-2}(t)$ est que :

$$\lim_{t \rightarrow \infty} \frac{H_{k+1}^{(1)} \cdot [H_{k-1}^{(2)}]^2}{[H_k^{(2)}]^2 \cdot H_{k-1}^{(1)}} = 0$$

démonstration : d'après (IV-8) on a :

$$\frac{\varepsilon'_{2k}(t)}{\varepsilon'_{2k-2}(t)} = \frac{H_k^{(1)} \cdot H_{k+1}^{(1)} \cdot [H_{k-1}^{(2)}]^2}{[H_k^{(2)}]^2 \cdot H_{k-1}^{(1)} \cdot H_k^{(1)}}$$

d'où la condition annoncée.

4.1.3 Introduction d'un facteur d'accélération

Supposons que la condition du théorème IV-4 ne soit pas remplie.

Alors $\varepsilon_{2k}(t)$ ne converge pas plus vite que $\varepsilon_{2k-2}(t)$. Nous définirons alors $\bar{\varepsilon}_{2k}(\omega_k, t)$ par :

$$\bar{\varepsilon}_{2k}(\omega_k, t) = \bar{\varepsilon}_{2k-2}(\omega_{k-1}, t) - \omega_k \frac{[H_k^{(1)}]^2}{H_{k-1}^{(2)} \cdot H_k^{(2)}} \quad (IV-9)$$

$$\varepsilon_0(\omega_0, t) = \langle x, x'(t) \rangle = f(t)$$

où les ω_k sont des facteurs d'accélération indépendants de t . La condition de convergence donnée par le théorème IV-2 s'applique aussi à $\bar{\epsilon}_{2k}(\omega_k, t)$. Il n'y a donc pas lieu de reformuler cette condition. Cependant on peut trouver une condition analogue à celle du théorème III-5 pour le choix de ω_k .

Théorème IV-5

Une condition nécessaire et suffisante pour que $\bar{\epsilon}_{2k}(\omega_k, t)$ converge vers S plus vite que $\bar{\epsilon}_{2k}(1, t)$ est de prendre :

$$\omega_k = \lim_{t \rightarrow \infty} \frac{1}{1 - \frac{\bar{\epsilon}'_{2k}(1, t)}{\bar{\epsilon}'_{2k-2}(\omega_{k-1}, t)}}$$

démonstration : Posons $R_k = \frac{[H_k^{(1)}]^2}{H_{k-1}^{(2)} \cdot H_k^{(2)}}$. Nous avons

$$\bar{\epsilon}'_{2k}(1, t) = \bar{\epsilon}'_{2k-2}(\omega_{k-1}, t) - R_k \quad \text{d'où}$$

$$\bar{\epsilon}'_{2k}(\omega_k, t) = \bar{\epsilon}'_{2k-2}(\omega_{k-1}, t) - \omega_k \left[\bar{\epsilon}'_{2k-2}(\omega_{k-1}, t) - \bar{\epsilon}'_{2k}(1, t) \right]$$

On obtient immédiatement la condition du théorème IV-5 en imposant que

$$\lim_{t \rightarrow \infty} \frac{\bar{\epsilon}'_{2k}(\omega_k, t)}{\bar{\epsilon}'_{2k}(1, t)} = 0. \quad \text{On voit que si } \lim_{t \rightarrow \infty} \frac{\bar{\epsilon}'_{2k}(1, t)}{\bar{\epsilon}'_{2k-2}(\omega_{k-1}, t)} = 0 \text{ alors}$$

la condition du théorème IV-4 est remplie et que l'on retrouve bien $\omega_k=1$.

Prenons, par exemple, $f(t) = 1 + \frac{1}{t}$. On trouve $\omega_1 = 2$, d'où l'algorithme :

$$\bar{\epsilon}'_2(2, t) = f(t) - 2 \frac{f'(t)}{f''(t)}$$

qui n'est autre que la première forme confluyente du φ -algorithme que l'on obtient par une procédure analogue à celle utilisée pour l' ϵ -algorithme (62). On arrive donc à la même conclusion que dans le cas discret, à savoir

que le facteur d'accélération ω_k apparait comme le lien entre les formes confluentes des ϵ et φ -algorithmes. Nous pouvons donc définir une classe d'algorithmes de la forme de $\bar{\epsilon}_{2k}(\omega_k, t)$. Le choix de ω_k donné par le théorème IV-5 nous fournira alors l'algorithme le plus rapidement convergent de la classe. Si nous connaissons analytiquement $f(t)$ et ses dérivées successives alors il sera possible de calculer la valeur exacte de ω_k .

Essayons maintenant de caractériser la fonction $\varphi(t, a)$ attachée à cette classe d'algorithmes.

Lemme IV-4

Une condition nécessaire et suffisante pour que $\bar{\epsilon}_{2k}(\omega_k, t)$ soit exacte pour $t \geq t_0$ est que $f(t)$ vérifie l'équation différentielle suivante $\forall t \geq t_0$:

$$S = \sum_{i=0}^k A_i \frac{\begin{vmatrix} f & \dots & f^{(i)} \\ \dots & \dots & \dots \\ f^{(i)} & \dots & f^{(2i)} \end{vmatrix}}{\begin{vmatrix} f'' & \dots & f^{(i+1)} \\ \dots & \dots & \dots \\ f^{(i+1)} & \dots & f^{(2i)} \end{vmatrix}} \quad \text{avec} \quad \sum_{i=0}^k A_i = 1$$

démonstration : on a :

$$\begin{aligned} \bar{\epsilon}_{2k}(\omega_k, t) &= \bar{\epsilon}_{2k-2}(\omega_{k-1}, t) - \omega_k R_k \\ &= \bar{\epsilon}_{2k-4}(\omega_{k-2}, t) - \omega_{k-1} R_{k-1} - \omega_k R_k \\ &= f - \sum_{i=1}^k \omega_i R_i \\ &= f + \sum_{i=1}^k \omega_i [\epsilon_{2i}(t) - \epsilon_{2i-1}(t)] \end{aligned}$$

posons $\omega_0 = 1$

$$A_i = \omega_i - \omega_{i+1} \quad i=0, \dots, k-1$$

$$A_k = \omega_k$$

on a :
$$\bar{\mathcal{E}}_{2k}(\omega_k, t) = \sum_{i=0}^k A_i \mathcal{E}_{2i}(t) \quad \text{et} \quad \sum_{i=0}^k A_i = 1$$

d'où l'équation différentielle du lemme IV-4 d'après la première relation (IV-5). Cette équation différentielle est difficile à résoudre dans le cas général. Nous l'avons résolue pour $k=1$.

Théorème IV-6

Une condition nécessaire et suffisante pour que $\bar{\mathcal{E}}_2(\omega_1, t)$ soit exacte pour $t \geq t_0$ est que $f(t) = c_1 e^{-c_2 t} + S$ (avec $c_2 > 0$) ou que :

$$f(t) = \left[(1-\omega_1) c_1 t + c_2 \right]^{1/(1-\omega_1)} + S \quad \text{pour } t \geq t_0$$

démonstration : La condition est nécessaire. En effet on doit avoir pour

$t \geq t_0$:

$$S = f - \omega_1 \frac{f',2}{f''} \quad \text{ou encore} \quad g - \omega_1 \frac{g',2}{g''} = 0$$

ce qui peut s'écrire $\frac{g}{g'} = \omega_1 \frac{g'}{g''}$ ou $\frac{g'}{g} = \frac{1}{\omega_1} \frac{g''}{g'}$

soit en intégrant :

$$\log g = \frac{1}{\omega_1} \log g' + c$$

d'où
$$g(t) = c \left[g'(t) \right]^{1/\omega_1}$$

ce qui donne $g^{-\omega_1} dg = c$

si $\omega_1 = 1$ on a $g(t) = C_1 e^{-c_2 t}$

si $\omega_1 \neq 1$ on a $g(t) = \left[(1-\omega_1) C_1 t + c_2 \right]^{1/(1-\omega_1)}$

La condition suffisante se démontre immédiatement en portant ces fonctions $f(t)$ dans (IV-9). En effet, si $f(t) = S + C_1 e^{-c_2 t}$ on trouve $\omega_1 = 1$ et par conséquent $\bar{\varepsilon}_2(1, t) = \varepsilon_2(t) = S$ pour $t \geq t_0$. Si $f(t) = S + [a C_1 t + C_2]$ on trouve $\omega_1 = 1-a$ et $\bar{\varepsilon}_2(1-a, t) = S$.

Il faut remarquer que cette dernière forme englobe, en posant $a = 1/n$ pour n entier, les fractions rationnelles dont numérateur et dénominateur sont des polynômes en t , le degré du numérateur étant inférieur ou égal à celui du dénominateur.

4-1-4 Introduction d'une fonction d'accélération

Nous allons maintenant essayer d'élargir encore la classe des fonctions pour lesquelles $\bar{\varepsilon}_{2k}(\omega_k, t)$ est exacte. C'est pour cette raison que nous introduisons un paramètre d'accélération dépendant de t : c'est la fonction d'accélération $\omega_k(t)$.

d'où un nouvel algorithme :

$$\bar{\varepsilon}_{2k}(\omega_k(t), t) = \bar{\varepsilon}_{2k-2}(\omega_{k-1}(t), t) - \omega_k(t) \cdot R_k \quad (\text{IV-10})$$

$$\text{avec } \bar{\varepsilon}_0(\omega_0(t), t) = \langle x, x'(t) \rangle = f(t).$$

Pour simplifier les notations nous écrirons dans la suite $\bar{\omega}_k$ quand il s'agira de la fonction d'accélération $\omega_k(t)$ et ω_k quand il s'agira du facteur d'accélération indépendant de t . Nous avons les résultats suivants :

Théorème IV-7

Supposons que $\lim_{t \rightarrow \infty} \bar{\varepsilon}_{2k-2}(\bar{\omega}_{k-1}, t) = S$. Une condition nécessaire et suffisante pour que $\lim_{t \rightarrow \infty} \bar{\varepsilon}_{2k}(\bar{\omega}_k, t) = S$ est que :

$$\lim_{t \rightarrow \infty} \omega_k(t) \frac{\left[H_k^{(1)} \right]^2}{H_{k-1}^{(2)} \cdot H_k^{(2)}} = 0$$

Lemme IV-5

Une condition nécessaire et suffisante pour que $\mathcal{E}_{2k}(\bar{\omega}_k, t)$ soit exacte pour $t \geq t_0$ est que $f(t)$ vérifie l'équation différentielle suivante $\forall t \geq t_0$:

$$S = \sum_{i=0}^k A_i(t) \frac{\begin{vmatrix} f & \dots & f^{(i)} \\ \dots & \dots & \dots \\ f^{(i)} & \dots & f^{(2i)} \end{vmatrix}}{\begin{vmatrix} f'' & \dots & f^{(i+1)} \\ \dots & \dots & \dots \\ f^{(i+1)} & \dots & f^{(2i)} \end{vmatrix}} \quad \text{avec} \quad \sum_{i=0}^k A_i(t) = 1 \quad \forall t$$

La démonstration est identique à celle du lemme IV-4.

Cette équation différentielle est difficile à résoudre dans le cas général.

Nous l'avons résolue pour $k = 1$.

Théorème IV-5

Une condition nécessaire et suffisante pour que $\mathcal{E}_2(\bar{\omega}_1, t)$ soit exacte

pour $t \geq t_0$ est que $f(t) = c_1 \exp \int \frac{dt}{c_2 + t - \Omega_1(t)} + S \quad \forall t \geq t_0, \Omega_1(t)$

désignant une primitive de $\omega_1(t)$.

démonstration: la condition est nécessaire. En effet, on doit avoir pour

$t \geq t_0$

$$S = f - \bar{\omega}_1 \frac{f,2}{f''} \quad \text{ou encore} \quad \bar{\omega}_1 \frac{g'}{g} = \frac{g''}{g}$$

cherchons les solutions de la forme $g(t) = e^{z(t)}$. On obtient pour $z(t)$

l'équation différentielle :

$$\bar{\omega}_1 - 1 = \frac{z''}{z'^2}$$

d'où

$-\frac{1}{z'} = -c_2 - t + \Omega_1(t)$ avec $\Omega_1(t)$ primitive de $\omega_1(t)$. En intégrant une nouvelle fois on obtient

$$z(t) = \int \frac{dt}{c_2 + t - \Omega_1(t)}$$

on démontre immédiatement que la condition est suffisante en portant la fonction $f(t)$ ainsi obtenue dans l'équation différentielle.

Pour $\bar{\mathcal{E}}_2(\bar{\omega}_1, t)$ nous avons également obtenu le résultat suivant :

Théorème IV-9

Une condition nécessaire et suffisante pour que $\bar{\mathcal{E}}_2(\bar{\omega}_1, t)$ soit exacte pour $t \geq t_0$ est que la condition du théorème IV-7 soit remplie et que $\omega_1(t)$ vérifie :

$$f''^2 - \bar{\omega}_1' f' f'' - \bar{\omega}_1 (2f''^2 - f' f''') = 0 \text{ pour } t \geq t_0$$

démonstration: la condition est nécessaire. Si $\bar{\mathcal{E}}_2(\bar{\omega}_1, t) = S$ alors

$$\bar{\mathcal{E}}_2'(\bar{\omega}_1, t) = 0 \text{ d'où :}$$

$$\bar{\mathcal{E}}_2'(\bar{\omega}_1, t) = f' - \bar{\omega}_1' \frac{f'^2}{f''} - \bar{\omega}_1 \frac{2f' f''^2 - f'^2 f'''}{f''^2} = 0 \text{ et}$$

$$f''^2 - \bar{\omega}_1' f' f'' - \bar{\omega}_1 (2f''^2 - f' f''') = 0$$

La condition est suffisante. Si cette équation différentielle est vérifiée on a alors $\bar{\mathcal{E}}_2'(\bar{\omega}_1, t) = 0$ et par conséquent $\bar{\mathcal{E}}_2(\bar{\omega}_1, t) = A$. Or, si la condition du théorème IV-7 est vérifiée, $\lim_{t \rightarrow \infty} \bar{\mathcal{E}}_2(\bar{\omega}_1, t) = S$ ce qui entraîne $A = S$. Il faut remarquer que l'équation différentielle du théorème IV-9 peut s'intégrer et nous donne $\omega_1(t) = \left[f + c_1 \right] \frac{f''}{f'^2}$ avec $c_1 = S$. Elle ne nous permet donc pas de calculer $\omega_1(t)$ pour $f(t)$ donnée car ce calcul nécessite la connaissance de S .

Nous avons enfin le résultat suivant :

Théorème IV-10

Une condition nécessaire et suffisante pour que $\bar{\varepsilon}_{2k}(\bar{\omega}_k, t)$ converge vers S plus vite que $\bar{\varepsilon}_{2k}(\omega_k, t)$ est que :

$$\lim_{t \rightarrow \infty} \frac{\bar{\varepsilon}'_{2k-2}(\bar{\omega}_{k-1}, t) - \bar{\omega}'_k R'_k - \bar{\omega}_k R'_k}{\bar{\varepsilon}'_{2k-2}(\omega_{k-1}, t) - \omega_k \cdot R'_k} = 0$$

La démonstration de ce théorème est immédiate à partir des définitions des deux algorithmes. Un cas particulièrement intéressant et simple est celui de $k = 1$.

On obtient alors comme condition d'accélération de la convergence :

$$\lim_{t \rightarrow \infty} \frac{f''^2 - \bar{\omega}'_1 f' f'' - \bar{\omega}_1 (2f''^2 - f' f'''')}{f''^2 - \omega_1 (2f''^2 - f' f'''')} = 0$$

on voit que si $\bar{\omega}'_1 = \omega_1$ ce rapport vaut 1 ce qui est normal.

4-2 Seconde forme confluyente de l' ε -algorithme

Dans la forme habituelle (II-5) de l' ε -algorithme remplaçons n par $t-a+n\Delta t$, $\varepsilon_{2k+1}^{(n)}$ par $\varepsilon_{2k+1}^*(t) / \Delta t$, $\varepsilon_{2k}^{(n)}$ par $\Delta t \cdot \varepsilon_{2k}^*(t)$ et faisons tendre Δt vers 0. On obtient la seconde forme confluyente de l' ε -algorithme (73) :

$$\varepsilon_{2k+1}^*(t) = \varepsilon_{2k-1}^*(t) + \frac{1}{\varepsilon_{2k}^*(t) + f(t)} \tag{IV-11}$$

$$\varepsilon_{2k+2}^*(t) = \varepsilon_{2k}^*(t) + \frac{1}{\varepsilon_{2k+1}^*(t)}$$

avec $\mathcal{E}_{-1}^*(t) = 0$ et $\mathcal{E}_0^*(t) = 0$

avec la convention :

$$H_k^{(-1)} = \begin{vmatrix} 0 & f & f' & \dots & f^{(k-2)} \\ f & f' & f'' & \dots & f^{(k-1)} \\ \dots & \dots & \dots & \dots & \dots \\ f^{(k-2)} & f^{(k-1)} & f^{(k)} & \dots & f^{(2k-3)} \end{vmatrix}$$

On peut écrire :

$$\mathcal{E}_{2k}^*(t) = \frac{H_{k+1}^{(-1)}}{H_k^{(1)}} \quad \mathcal{E}_{2k+1}^*(t) = \frac{H_k^{(2)}}{H_{k+1}^{(0)}} \tag{IV-12}$$

$$\mathcal{E}_{2k+2}^*(t) = \mathcal{E}_{2k}^*(t) - \frac{[H_{k+1}^{(0)}]^2}{H_k^{(1)} \cdot H_{k+1}^{(1)}}$$

Wynn (63) a établi des relations qui lient les deux formes confluentes de l' \mathcal{E} -algorithme. Nous appellerons respectivement $\mathcal{E}_k^{(n)}(t)$ et $\mathcal{E}_k^{(n)*}(t)$ les quantités obtenues par applications des première et seconde formes confluentes de l' \mathcal{E} -algorithme à $f(t) = g^{(n)}(t)$. Si nous introduisons de plus la notation :

$$g^{(-1)}(t) = \int_a^t g(x) dx$$

nous avons :

$$\begin{aligned} \mathcal{E}_{2k+1}^{(n)*}(t) &= \mathcal{E}_{2k+1}^{(n-1)}(t) \\ \mathcal{E}_{2k}^{(n)*}(t) &= \mathcal{E}_{2k}^{(n-1)}(t) - g^{(n-1)}(t) \\ &= \frac{1}{\mathcal{E}_{2k+1}^{(n-2)}(t)} - g^{(n-1)}(t) \\ &= \frac{1}{\mathcal{E}_{2k+1}^{(n-1)*}(t)} - g^{(n-1)}(t) \end{aligned}$$

Wynn a montré (73) que, sous certaines conditions, on avait

$$\varepsilon_{2k}^*(t) = \int_t^{\infty} f(\tau) d\tau \quad \text{avec } f(\tau) = \langle x, x(\tau) \rangle . \text{ Ce sont}$$

ces conditions que nous allons maintenant étudier.

4.2.1. Exactitude de $\varepsilon_{2k}^*(t)$

Lemme IV-6

Une condition nécessaire et suffisante pour que $\varepsilon_{2k}^*(t) = \int_t^{\infty} f(x) dx$ est

que :

$$\left| \begin{array}{cccc} - \int_t^{\infty} f(x) dx & f & f' & \dots & f^{(k-1)} \\ & f & f' & f'' & \dots & f^{(k)} \\ \hline & f^{(k-1)} & f^{(k)} & f^{(k+1)} & \dots & f^{(2k-1)} \end{array} \right| = 0$$

la démonstration est identique à celle du lemme IV-1. Pour cette seconde forme confluente de l' ε -algorithme nous dirons donc que $\varepsilon_{2k}^*(t)$ est exacte pour $t \geq t_0$ si $\varepsilon_{2k}^*(t) = \int_t^{\infty} f(x) dx \quad \forall t \geq t_0$.

Lemme IV-7

Une condition nécessaire et suffisante pour que $\varepsilon_{2k}^*(t)$ soit exacte pour $t \geq t_0$ est que $g(t) = \int_t^{\infty} f(x) dx$ vérifie une équation différentielle linéaire d'ordre k à coefficients constants $\forall t \geq t_0$.

Corollaire IV-1 :

Une condition nécessaire et suffisante pour que $\varepsilon_{2k}^*(t)$ soit exacte pour $t \geq t_0$ est que $f(t)$ vérifie une équation différentielle linéaire d'ordre k à coefficients constants $\forall t \geq t_0$. La démonstration du lemme IV-7 est identique à celle du lemme IV-2. La démonstration du corollaire IV-1 est évidente. Il

en résulte que la condition nécessaire et suffisante d'exactitude du théorème IV-1 est encore valable pour $\varepsilon_{2k}^*(t)$

4.2.2 Convergence de $\varepsilon_{2k}^*(t)$

Théorème IV-11

Supposons que $\lim_{t \rightarrow \infty} \varepsilon_{2k-2}^*(t) = 0$. Une condition nécessaire et suffisante pour que $\lim_{t \rightarrow \infty} \varepsilon_{2k}^*(t) = 0$ est que :

$$\lim_{t \rightarrow \infty} \frac{[H_k^{(0)}]^2}{H_{k-1}^{(1)} \cdot H_k^{(1)}} = 0$$

La démonstration de ce théorème est immédiate. Elle découle de la troisième des relations (IV-12).

Lemme IV-8

On a les relations suivantes:

$$\varepsilon_{2k+1}^*(t) = - \frac{H_k^{(1)} \cdot H_{k+1}^{(1)}}{[H_{k+1}^{(0)}]^2} \quad (\text{IV-13})$$

$$\varepsilon_{2k}^*(t) = \frac{H_k^{(0)} \cdot H_{k+1}^{(0)}}{[H_k^{(1)}]^2} - f(t) \quad (\text{IV-14})$$

La démonstration de ce lemme est analogue à celle du lemme IV-3.

Théorème IV-12

Une condition nécessaire et suffisante pour que $\varepsilon_{2k}^*(t)$ tende vers 0 plus vite que $\varepsilon_{2k-2}^*(t)$ est que :

$$\lim_{t \rightarrow \infty} \frac{H_{k+1}^{(0)} \cdot [H_{k-1}^{(1)}]^2}{[H_k^{(1)}]^2 \cdot H_{k-1}^{(0)}} = 0$$

La démonstration est identique à celle du théorème IV-4 et est une conséquence directe du lemme IV-8

4.2.3 Introduction d'un facteur d'accélération

Supposons que la condition du théorème IV-12 ne soit remplie.

Alors $\varepsilon_{2k}^*(t)$ ne tendra pas plus vite que $\varepsilon_{2k-2}^*(t)$ vers 0. Nous définirons alors $\bar{\varepsilon}_{2k}^*(\omega_k, t)$ par :

$$\bar{\varepsilon}_{2k}^*(\omega_k, t) = \bar{\varepsilon}_{2k-2}^*(\omega_{k-1}, t) - \omega_k \frac{[H_k^{(0)}]^2}{H_{k-1}^{(1)} \circ H_k^{(1)}} \quad (\text{IV-15})$$

avec $\bar{\varepsilon}_{-1}^*(t) = 0$ et $\bar{\varepsilon}_0^*(\omega_0, t) = 0$

ω_k est un facteur d'accélération indépendant de t . La condition de convergence du théorème IV-11 est donc encore valable pour $\bar{\varepsilon}_{2k}^*(\omega_k, t)$. Nous avons :

Théorème IV-13

Une condition nécessaire et suffisante pour que $\bar{\varepsilon}_{2k}^*(\omega_k, t)$ tende vers 0 plus vite que $\bar{\varepsilon}_{2k}^*(1, t)$ est que :

$$\omega_k = \lim_{t \rightarrow \infty} \frac{1}{1 - \frac{\bar{\varepsilon}_{2k}^*(1, t)}{\bar{\varepsilon}_{2k-2}^*(\omega_{k-1}, t)}}$$

La démonstration est identique à celle du théorème IV-5.

Pour cette seconde forme confluyente de l' ε -algorithme on arrive, par conséquent, aux mêmes conclusions que pour la première forme confluyente. On peut donc définir une classe d'algorithmes de la forme (IV-15). Le choix de ω_k donné par le théorème IV-13 nous fournit alors l'algorithme de la classe qui converge le plus rapidement. Si la condition d'accélération de la convergence du théorème IV-12 est vérifiée, il sera impossible d'accélérer encore cette convergence.

Essayons maintenant de caractériser la fonction $\Psi(t, a)$ attachée à cette classe d'algorithmes.

Lemme IV-9

Une condition nécessaire et suffisante pour que $\overline{\varepsilon}_{2k}^*(\omega_k, t)$ soit exacte pour $t \geq t_0$ est que $f(t)$ vérifie l'équation différentielle suivante ($\forall t \geq t_0$)

$$\int_t^\infty f(x) dx = \sum_{i=0}^k A_i \frac{\begin{vmatrix} 0 & f & \dots & f^{(i-1)} \\ \dots & \dots & \dots & \dots \\ f^{(i-1)} & f^{(i)} & \dots & f^{(2i-1)} \end{vmatrix}}{\begin{vmatrix} f' & \dots & f^{(i)} \\ \dots & \dots & \dots \\ f^{(i)} & \dots & f^{(2i-1)} \end{vmatrix}} \quad \text{avec} \quad \sum_{i=0}^k A_i = 1$$

La démonstration est identique à celle du lemme IV-4. Nous avons résolu cette équation intégro-différentielle pour $k=1$

Théorème IV-14 -

Une condition nécessaire et suffisante pour que $\overline{\varepsilon}_2^*(\omega_1, t)$ soit exacte pour $t \geq t_0$ est que $f(t) = C_1 e^{-c_2 t}$ (avec $c_2 > 0$) ou que $f(t) = 0$ ou que $f(t) = \left[\frac{\omega_1 - 1}{2\omega_1 - 1} c_1 t + c_2 \right]^{1/(2\omega_1 - 1)} / (\omega_1 - 1)$ pour $t \geq t_0$

démonstration : La condition est nécessaire. En effet on doit avoir pour $t \geq t_0$:

$$-\omega_1 \frac{f^2}{f'} = \int_t^\infty f(x) dx$$

en dérivant on obtient :

$$\frac{f}{f'} = \frac{2\omega_1 - 1}{\omega_1} \cdot \frac{f''}{f}$$

Soit en intégrant :

$$f = C \left[f' \right]^{\frac{2\omega_1-1}{\omega_1}}$$

si $\omega_1=1$ on trouve $f=c_1 e^{-c_2 t}$

si $\omega_1 = \frac{1}{2}$ on trouve $f = 0$

si $\omega_1 \neq 1, \frac{1}{2}$ on trouve $f = \left[\frac{\omega_1-1}{2\omega_1-1} c_1 t + c_2 \right]^{\frac{2\omega_1-1}{\omega_1-1}}$

La démonstration de la condition suffisante est immédiate.

4.2.4. Introduction d'une fonction d'accélération

A la place d'un paramètre d'accélération indépendant de t , nous allons maintenant introduire une fonction d'accélération $\omega_k(t)$ (notée $\overline{\omega}_k$). D'où un nouvel algorithme.

$$\overline{\varepsilon}_{2k}^*(\omega_k(t), t) = \overline{\varepsilon}_{2k-2}^*(\omega_{k-1}(t), t) - \omega_k(t). \quad R_k \quad (IV-16)$$

$$\overline{\varepsilon}_0^*(\omega_0(t), t) = 0 \quad R_k = \frac{[H_k^{(0)}]^2}{H_{k-1}^{(1)} \cdot H_k^{(1)}}$$

Théorème IV-15

Supposons que $\lim_{t \rightarrow \infty} \overline{\varepsilon}_{2k-2}^*(\overline{\omega}_{k-1}, t) = 0$. Une condition nécessaire et suffisante

pour que $\lim_{t \rightarrow \infty} \overline{\varepsilon}_{2k}^*(\overline{\omega}_k, t) = 0$ est que :

$$\lim_{t \rightarrow \infty} \omega_k(t) \frac{[H_k^{(0)}]^2}{H_{k-1}^{(1)} \cdot H_k^{(1)}} = 0$$

Lemme IV-10

Une condition nécessaire et suffisante pour que $\overline{\varepsilon}_{2k}^*(\overline{\omega}_k, t)$ soit exacte pour $t \geq t_0$ est que $f(t)$ vérifie l'équation intégral-différentielle suivante $\forall t \geq t_0$:

$$\int_t^{\infty} f(x) dx = \sum_{i=0}^k A_i(t) \frac{\begin{vmatrix} 0 & f & \dots & f^{(i-1)} \\ \dots & \dots & \dots & \dots \\ f^{(i-1)} & f^{(i)} & \dots & f^{(2i-1)} \\ \dots & \dots & \dots & \dots \\ f' & \dots & \dots & f^{(i)} \\ \dots & \dots & \dots & \dots \\ f^{(i)} & \dots & \dots & f^{(2i-1)} \end{vmatrix}}{\begin{vmatrix} f' & \dots & \dots & f^{(i)} \\ \dots & \dots & \dots & \dots \\ f^{(i)} & \dots & \dots & f^{(2i-1)} \end{vmatrix}} \text{ avec } \sum_{i=0}^k A_i(t) = 1$$

La démonstration est identique à celle du lemme IV-9. Nous avons résolu cette équation pour $k=1$.

Théorème IV-16

Une condition nécessaire et suffisante pour que $\bar{\varepsilon}_2^*(\bar{\omega}_1, t)$ soit exacte pour $t \geq t_0$ est que $f(t) = c_1 \exp \left[\int \frac{dt}{c_2 + t - \Omega_1(t)} \right]$ pour $t \geq t_0$. $\Omega_1(t)$ désigne une primitive de $\frac{2\omega_1(t)-1}{\omega_1(t)}$.

La démonstration est identique à celle du théorème IV-8. On a également les deux résultats suivants :

Théorème IV-17

Une condition nécessaire et suffisante pour que $\bar{\varepsilon}^*(\bar{\omega}_1, t)$ soit exacte pour $t \geq t_0$ est que la condition du théorème IV-15 soit remplie et que $\omega_1(t)$ vérifie l'équation différentielle suivante $\forall t \geq t_0$:

$$f'^2 - \bar{\omega}'_1 f f' - \bar{\omega}_1 (2f'^2 - f f'') = 0$$

la démonstration est identique à celle du théorème IV-9

Théorème IV-18

Une condition nécessaire et suffisante pour que $\bar{\varepsilon}_{2k}^*(\bar{\omega}_k, t)$ tende vers 0 plus vite que $\bar{\varepsilon}_{2k}^*(\omega_k, t)$ est que :

$$\lim_{t \rightarrow \infty} \frac{\varepsilon_{2k-2}^{*'}(\bar{\omega}_{k-1}, t) - \bar{\omega}'_k R_k - \bar{\omega}_k R'_k}{\varepsilon_{2k-2}^{*'}(\omega_{k-1}, t) - \omega_k R'_k} = 0$$

La démonstration découle immédiatement de la définition des deux algorithmes. Pour $k=1$ cette condition prend la forme simple :

$$\lim_{t \rightarrow \infty} \frac{\bar{\omega}'_1 f f' + \bar{\omega}_1 (2f'^2 - f f'')}{\omega_1 (2f'^2 - f f'')} = 0$$

on voit que si $\bar{\omega}_1 = \omega_1$ ce rapport vaut 1 ce qui est normal.

4-3 Formes confluentes et développements asymptotiques

Tous les résultats que nous venons d'établir au cours de ce chapitre s'expliquent facilement si l'on considère les $\omega_k R_k$ comme les termes successifs d'un développement asymptotique. Soit $f(t)$ une fonction donnée de t , telle que $\lim_{t \rightarrow \infty} f(t) = S$. Soit G l'ensemble des fonctions $R_i(t)$ définies par :

$$R_i(t) = \frac{[H_i^{(1)}]^2}{H_{i-1}^{(2)} \circ H_i^{(2)}}$$

supposons que $R_{i+1}(t) = o(R_i(t))$ ce qui nécessite, notons le au passage, que $R_i = o(1) \forall i$ qui n'est autre que la condition du théorème IV-2. Alors, s'il vérifie cette propriété, l'ensemble G est une échelle de comparaison (7,20). Cherchons les développement asymptotique de $g(t) = f(t) - S$ par rapport à cette échelle au voisinage de $+\infty$. Si un tel développement existe jusqu'à l'ordre n on aura :

$$g(t) = \sum_{i=1}^n \omega_i R_i(t) + o(R_n(t))$$

Il faut donc trouver les constantes ω_i telles que :

$$g(t) = \sum_{i=1}^k \omega_i R_i(t) + o(R_k) \quad k = 1, 2, \dots, n$$

Posons $g_k = \sum_{i=1}^k \omega_i R_i(t)$. Nous avons

$$f(t) - S = g_k + o(R_k)$$

$$S = f(t) - g_k + o(R_k)$$

$$= \bar{E}_{2k}(\omega_k, t) + o(R_k)$$

$$= \bar{E}_{2k-2}(\omega_{k-1}, t) - \omega_k R_k + o(R_k)$$

on va choisir ω_k de façon à avoir :

$$o = \bar{E}'_{2k-2}(\omega_{k-1}, t) - \omega_k R'_k + o(R'_k)$$

ce qui entraînera $S = \bar{E}'_{2k}(\omega_k, t) = o(R_k)$ en supposant, ce qui est vérifié, que

$\int_t^\infty \bar{E}'_{2k-2}(\omega_{k-1}, x) dx$ et $\int_t^\infty R'_k(x) dx$ sont convergentes. On a donc :

$$\omega_k R'_k = \bar{E}'_{2k-2}(\omega_{k-1}, t) + o(R'_k)$$

d'où :

$$\omega_k = \lim_{t \rightarrow \infty} \frac{\bar{E}'_{2k-2}(\omega_{k-1}, t)}{R'_k}$$

or : $R'_k = \bar{E}'_{2k-2}(\omega_{k-1}, t) - \bar{E}'_{2k}(1, t)$ ce qui donne :

$$\omega_k = \lim_{t \rightarrow \infty} \frac{1}{1 - \frac{\bar{E}'_{2k}(1, t)}{\bar{E}'_{2k-2}(\omega_{k-1}, t)}}$$

ce qui n'est autre que la condition du théorème IV-5. D'un autre côté le théorème IV-4 apparaît comme une condition nécessaire et suffisante pour que $\omega_k = 1$. Le fait que le choix de ω_k donné par le théorème IV-5 fournisse le plus

rapide de tous les algorithmes de la forme (IV-9) signifie simplement que ω_k est le coefficient de R_k dans le développement asymptotique de $g(t)$ par rapport à G au voisinage de $+\infty$. Donc ce choix de ω_k donne le seul algorithme de la classe pour lequel on ait $\bar{\mathcal{E}}_{2k}(\omega_k, t) - S = o(R_k)$. Les autres choix de ω_k conduisent à $O(R_k)$ au lieu de $o(R_k)$. De même l'introduction d'une fonction d'accélération dépendant de t apparaît comme un développement asymptotique généralisé et l'on retrouve les théorèmes correspondants.

De cette comparaison des formes confluentes et des développements asymptotiques on peut déduire trois conséquences pratiques :

1°) On peut obtenir le développement asymptotique autour de n'importe quel point t_0 par un changement de variable $\tau = t_0 \pm \frac{1}{t}$.

2°) Nous avons $S - \bar{\mathcal{E}}_{2k-2}(\omega_{k-1}, t) = -\omega_k R_k + o(R_k)$ donc :

$$S - \bar{\mathcal{E}}_{2k-2}(\omega_{k-1}, t) \sim -\omega_k R_k$$

et par conséquent :

$$S - \bar{\mathcal{E}}_{2k-2}(\omega_{k-1}, t) = O(R_k).$$

Donc si on arrête le développement asymptotique à l'indice $k-1$ l'erreur sera de l'ordre de grandeur de R_k . Inversement, si nous désirons estimer S avec une certaine précision, si nous nous fixons k cette estimation de l'erreur nous fournira la valeur de t à prendre et si t est fixée elle donnera la valeur de k .

3°) $\bar{\mathcal{E}}_{2k}(\omega_k, t)$ est une meilleure approximation de S que

$\bar{\mathcal{E}}_{2k-2}(\omega_{k-1}, t)$. Enfin il nous est possible, grâce aux formes confluentes de l' \mathcal{E} -algorithme, de retrouver un résultat connu sur la partie principale d'une primitive. En effet, si nous prenons $f(t) = \int_a^t g(x) dx$ nous aurons :

$$\mathcal{E}_2(t) = \int_a^t g(x) dx - \frac{g^2(t)}{g'(t)}$$

et par conséquent nous obtenons :

$$\int_t^{\infty} g(x) dx \sim \frac{g^2(t)}{g'(t)}$$

Avant de terminer ce chapitre signalons qu'il est inutile de recommencer cette étude pour les formes confluentes du \mathfrak{E} -algorithme puisque le paramètre d'accélération ω_k permet de faire la liaison entre les deux.

4-4 Théorèmes particuliers de convergence

Les théorèmes suivants sont une conséquence directe du théorème III-3.

Théorème IV-19

Si la suite $\{a_k = f^{(k)}(t)\}$ ou la suite $\{-a_k\}$ est soit totalement monotone, soit totalement oscillante, soit totalement oscillante décalée pour une certaine valeur de t et pour $k=0,1,\dots$, si $\lim_{t \rightarrow \infty} f(t) = S$ et si on applique la première forme confluyente de l' \mathfrak{E} -algorithme avec $\mathfrak{E}_0(t) = f(t)$ alors on a :

$$0 \leq |\mathfrak{E}_{2k+2}(t)| \leq |\mathfrak{E}_{2k}(t)| \quad \forall k$$

$$\text{et } \lim_{k \rightarrow \infty} \mathfrak{E}_{2k}(t) = S$$

Si la propriété de monotonie ou d'oscillation de la suite $\{a_k\}$ reste vraie quelque soit t alors :

$$\lim_{t \rightarrow \infty} \mathfrak{E}_{2k}(t) = S \quad \forall k$$

Théorème IV-20

Si la suite $\{a_0 = 0, a_k = f^{(k-1)}(t) \quad \forall k > 0\}$ ou la suite $\{-a_k\}$ est soit totalement monotone, soit totalement oscillante, soit totalement oscillante décalée pour une certaine valeur de t et si on applique la seconde forme confluyente de l' \mathfrak{E} -algorithme à $f(t)$ alors on a :

$$0 \leq \left| \varepsilon_{2k+2}^{\#}(t) \right| \leq \left| \varepsilon_{2k}^{\#}(t) \right| \quad \forall k$$

$$\text{et } \lim_{k \rightarrow \infty} \varepsilon_{2k}^{\#}(t) = 0$$

Si la propriété de monotonie ou d'oscillation de la suite $\{a_k\}$ reste vraie quelque soit t alors :

$$\lim_{t \rightarrow \infty} \varepsilon_{2k}^{\#}(t) = 0 \quad \forall k.$$

En pratique, seule sera utilisée la première forme confluente de l' ε -algorithme puisque c'est la plus simple et qu'elle est reliée à la seconde forme confluente par des relations données au paragraphe 4.2.

CHAPITRE V

PROGRAMMATION DES ALGORITHMES

Ce chapitre traite de la mise en oeuvre pratique des formes discrètes des algorithmes étudiés aux chapitres précédents. Après avoir établi les règles particulières à utiliser pour éviter la perte de précision due à la concellation, on parle de la stabilité et de la propagation des erreurs d'arrondis. On donne pour terminer les diverses possibilités d'emploi des algorithmes.

5-1. Considération théoriques

Soit un algorithme de la forme:

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

avec $\theta_{-1}^{(n)} = 0 \quad \forall n$ et $\theta_0^{(n)} = S_n$ donné $\forall n$.

Posons :
$$D_k^{(n)} = \frac{a_k^{(n)}}{\theta_k^{(n+1)} - \theta_k^{(n)}}$$

Un tel algorithme est un algorithme de losange. On place les quantités calculées par (V-1) dans une table à double entrée analogue à une table de différences. Comme nous l'avons vu au paragraphe 2.3.1 l'indice inférieur désigne la colonne et l'indice supérieur la diagonale descendante.

Nous allons étudier les règles particulières puis la stabilité d'un tel algorithme et enfin l'effet de lissage de l' ϵ -algorithme mis en évidence par Wynn (67).

5.1.1. Règles particulières

Supposons que les quantités $\theta_{k-2}^{(n+1)}$ et $\theta_{k-2}^{(n+2)}$ deviennent toutes les deux égales à b. Alors $\theta_{k-1}^{(n+1)}$ devient infini, $\theta_k^{(n)}$ et $\theta_k^{(n+1)}$ sont égaux à b et $\theta_{k+1}^{(n)}$ est indéterminé. La situation peut se résumer ainsi (70) :

$$\begin{array}{ccccccc}
 & & & & \theta_{k-1}^{(n)} & & \\
 & & & & & & \\
 & & & & & & \\
 \theta_{k-3}^{(n+2)} & & \theta_{k-2}^{(n+1)} = b & & & \theta_k^{(n)} = b & \\
 & & & & \theta_{k-1}^{(n+1)} = \infty & & \theta_{k+1}^{(n)} = ? \\
 & & & & & & \\
 & & \theta_{k-2}^{(n+2)} = b & & & \theta_k^{(n+1)} = b & \\
 & & & & \theta_{k-1}^{(n+2)} & &
 \end{array}$$

Si $\theta_{k-2}^{(n+1)}$ et $\theta_{k-2}^{(n+2)}$ sont très voisins et de mêmes signes il y a une perte de précision importante due à la cancellation. Alors $\theta_{k-1}^{(n+1)}$ est grand et mal déterminé d'où une grande imprécision dans la suite de l'application de l'algorithme. Pour remédier à ce fait on emploie des règles particulières que nous allons maintenant établir. On peut écrire :

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

$$\text{or } \theta_{k-2}^{(n+2)} - \theta_{k-2}^{(n+1)} = \frac{a_{k-2}^{(n+1)}}{\theta_{k-1}^{(n+1)} - \theta_{k-3}^{(n+2)}}$$

d'où

$$\theta_{k+1}^{(n)} = \theta_{k-1}^{(n+1)} + \frac{a_k^{(n)}}{\frac{a_{k-2}^{(n+1)}}{\theta_{k-1}^{(n+1)} - \theta_{k-3}^{(n+2)}} + \frac{a_{k-1}^{(n+1)}}{\theta_{k-1}^{(n+2)} - \theta_{k-1}^{(n+1)}} - \frac{a_{k-1}^{(n)}}{\theta_{k-1}^{(n+1)} - \theta_{k-1}^{(n)}}$$

$$\theta_{k+1}^{(n)} = \theta_{k-1}^{(n+1)} - \frac{a_k^{(n)} \theta_{k-1}^{(n+1)}}{a_{k-1}^{(n+1)} + a_{k-1}^{(n)} - a_{k-2}^{(n+1)} - \frac{\theta_{k-3}^{(n+2)} a_{k-2}^{(n+1)}}{\theta_{k-1}^{(n+1)} - \theta_{k-3}^{(n+2)}} + \frac{\theta_{k-1}^{(n+2)} a_{k-1}^{(n+1)}}{\theta_{k-1}^{(n+1)} - \theta_{k-1}^{(n+2)}} + \frac{\theta_{k-1}^{(n)} a_{k-1}^{(n)}}{\theta_{k-1}^{(n+1)} - \theta_{k-1}^{(n)}}$$

Posons

$$a = \left[a_{k-1}^{(n+1)} + a_{k-1}^{(n)} - a_{k-2}^{(n+1)} - a_k^{(n)} \right] \theta_{k-1}^{(n+1)} - a_{k-2}^{(n+1)} \theta_{k-1}^{(n+1)} \theta_{k-3}^{(n+2)} \left[\theta_{k-1}^{(n+1)} - \theta_{k-3}^{(n+2)} \right]^{-1} \\ + a_{k-1}^{(n+1)} \theta_{k-1}^{(n+1)} \theta_{k-1}^{(n+2)} \left[\theta_{k-1}^{(n+1)} - \theta_{k-1}^{(n+2)} \right]^{-1} + a_{k-1}^{(n)} \theta_{k-1}^{(n)} \theta_{k-1}^{(n+1)} \left[\theta_{k-1}^{(n+1)} - \theta_{k-1}^{(n)} \right]^{-1}$$

ou encore

$$\begin{aligned} a = & \left[a_{k-1}^{(n+1)} + a_{k-1}^{(n)} - a_{k-2}^{(n+1)} - a_k^{(n)} \right] \theta_{k-1}^{(n+1)} + a_{k-1}^{(n+1)} \theta_{k-1}^{(n+2)} \left[1 - \theta_{k-1}^{(n+2)} \cdot \theta_{k-1}^{(n+1)-1} \right]^{-1} \\ & + a_{k-1}^{(n)} \theta_{k-1}^{(n)} \left[1 - \theta_{k-1}^{(n)} \cdot \theta_{k-1}^{(n+1)-1} \right]^{-1} - a_{k-2}^{(n+1)} \theta_{k-3}^{(n+2)} \left[1 - \theta_{k-3}^{(n+2)} \cdot \theta_{k-1}^{(n+1)-1} \right]^{-1} \end{aligned}$$

d'où

$$\theta_{k+1}^{(n)} = a \left[a_k^{(n)} + \frac{a}{\theta_{k-1}^{(n+1)}} \right]^{-1}$$

Si $a_k^{(n)} = 1$ on retrouve l' \mathcal{E} -algorithme et ses règles particulières données par Wynn (70) :

$$\begin{aligned} a = & \mathcal{E}_{k-1}^{(n+2)} \left[1 - \mathcal{E}_{k-1}^{(n+2)} \cdot \mathcal{E}_{k-1}^{(n+1)-1} \right]^{-1} + \mathcal{E}_{k-1}^{(n)} \left[1 - \mathcal{E}_{k-1}^{(n)} \cdot \mathcal{E}_{k-1}^{(n+1)-1} \right]^{-1} \\ & - \mathcal{E}_{k-3}^{(n+2)} \left[1 - \mathcal{E}_{k-3}^{(n+2)} \cdot \mathcal{E}_{k-1}^{(n+1)-1} \right]^{-1} \\ \mathcal{E}_{k+1}^{(n)} = & a \left[1 + a \mathcal{E}_{k-1}^{(n+1)-1} \right]^{-1} \end{aligned}$$

Si $\mathcal{E}_{k-2}^{(n+1)}$ et $\mathcal{E}_{k-2}^{(n+2)}$ sont rigoureusement égaux la règle particulière se réduit à :

$$\mathcal{E}_{k+1}^{(n)} = \mathcal{E}_{k-1}^{(n+2)} + \mathcal{E}_{k-1}^{(n)} - \mathcal{E}_{k-3}^{(n+2)}$$

Pour $a_k^{(n)} = k+1$ on retrouve le \mathcal{P} -algorithme simplifié :

$$\begin{aligned} a = & k \mathcal{P}_{k-1}^{(n)} \left[1 - \mathcal{P}_{k-1}^{(n)} \cdot \mathcal{P}_{k-1}^{(n+1)-1} \right]^{-1} + k \mathcal{P}_{k-1}^{(n+2)} \left[1 - \mathcal{P}_{k-1}^{(n+2)} \cdot \mathcal{P}_{k-1}^{(n+1)-1} \right]^{-1} \\ & - (k-1) \mathcal{P}_{k-3}^{(n+2)} \left[1 - \mathcal{P}_{k-3}^{(n+2)} \cdot \mathcal{P}_{k-1}^{(n+1)-1} \right]^{-1} \end{aligned}$$

$$\mathcal{P}_{k+1}^{(n)} = a \left[k+1 + a \mathcal{P}_{k-1}^{(n+1)-1} \right]^{-1}$$

Si $\rho_{k-2}^{(n+1)}$ et $\rho_{k-2}^{(n+2)}$ sont rigoureusement égaux on a :

$$\rho_{k+1}^{(n)} = (k+1)^{-1} \left[k \cdot \rho_{k-1}^{(n)} + k \cdot \rho_{k-1}^{(n+2)} - (k-1) \cdot \rho_{k-3}^{(n+2)} \right]$$

5.1.2. Stabilité et propagation des erreurs

La stabilité et la propagation des erreurs ont également été étudiées par Wynn (67,68) dans le cas de l' ϵ -algorithme. Ses conclusions s'étendent facilement aux autres algorithmes que nous avons rencontré.

Supposons qu'à partir d'une valeur exacte $\epsilon_{k-1}^{(n)}$ on calcule les valeurs exactes $\epsilon_k^{(n)}$, $\epsilon_k^{(n-1)}$, etc... Introduisons maintenant une erreur relative $\delta_{k-1}^{(n)}$ sur $\epsilon_{k-1}^{(n)}$. Cette erreur engendre les erreurs relatives $\delta_k^{(n)}$, $\delta_k^{(n-1)}$, etc... respectivement sur $\epsilon_k^{(n)}$, $\epsilon_k^{(n-1)}$, ... La propagation de l'erreur s'effectue classiquement suivant un triangle :

$$\begin{array}{rcc} & & \epsilon_{k+1}^{(n-2)} (1 + \delta_{k+1}^{(n-2)}) \\ & & \epsilon_k^{(n-1)} (1 + \delta_k^{(n-1)}) \\ \epsilon_{k-1}^{(n)} (1 + \delta_{k-1}^{(n)}) & & \epsilon_{k+1}^{(n-1)} (1 + \delta_{k+1}^{(n-1)}) \\ & & \epsilon_k^{(n)} (1 + \delta_k^{(n)}) \\ & & \epsilon_{k+1}^{(n)} (1 + \delta_{k+1}^{(n)}) \end{array}$$

On a les relations suivantes :

$$\delta_k^{(n-1)} = - \frac{\epsilon_{k-1}^{(n)}}{\epsilon_k^{(n-1)} [\epsilon_{k-1}^{(n)} - \epsilon_{k-1}^{(n-1)}]^2} \delta_{k-1}^{(n)}$$

$$\delta_k^{(n)} = \frac{\epsilon_{k-1}^{(n)}}{\epsilon_k^{(n)} [\epsilon_{k-1}^{(n+1)} - \epsilon_{k-1}^{(n)}]^2} \delta_{k-1}^{(n)} \quad \text{etc.....}$$

Pour un algorithme de la forme (V-1) l'étude de l'influence d'erreurs sur les valeurs initiales $\theta_0^{(n)} = S_n$ s'effectue à l'aide des nombres de condition.

Supposons qu'il y ait une erreur δS_n sur S_n ; il y aura une erreur $\delta(\theta_k^{(n)}, S_n)$ sur $\theta_k^{(n)}$ telle que

$$\delta(\theta_k^{(n)}, S_n) = \frac{\partial \theta_k^{(n)}}{\partial S_n} \cdot \delta S_n$$

Une indication sur la stabilité numérique de l'algorithme est fournie par les nombres de condition $c(\theta_k^{(n)}, S_n)$ qui sont le rapport de l'erreur relative sur $\theta_k^{(n)}$ à celle sur S_n .

$$c(\theta_k^{(n)}, S_n) = \frac{S_n \cdot \delta \theta_k^{(n)}}{\theta_k^{(n)} \cdot \delta S_n}$$

Si le module d'un nombre de condition quelconque est supérieur à un, alors l'algorithme est instable. Pour l' \mathcal{E} -algorithme la propagation de l'erreur absolue est décrite par l'équation (68) :

$$(\mathcal{E}_{k+1}^{(n)} - \mathcal{E}_{k-1}^{(n+1)}) \left[\delta(\mathcal{E}_k^{(n+1)}, S_n) - \delta(\mathcal{E}_k^{(n)}, S_n) \right] + (\mathcal{E}_k^{(n+1)} - \mathcal{E}_k^{(n)}) \left[\delta(\mathcal{E}_{k+1}^{(n)}, S_n) - \delta(\mathcal{E}_{k-1}^{(n+1)}, S_n) \right] = 0$$

Les nombres de condition sont reliés par :

$$(\mathcal{E}_{k+1}^{(n)} - \mathcal{E}_{k-1}^{(n+1)}) \left[\mathcal{E}_k^{(n+1)} \cdot c(\mathcal{E}_k^{(n+1)}, S_n) - \mathcal{E}_k^{(n)} \cdot c(\mathcal{E}_k^{(n)}, S_n) \right] + (\mathcal{E}_k^{(n+1)} - \mathcal{E}_k^{(n)}) \left[\mathcal{E}_{k+1}^{(n)} \cdot c(\mathcal{E}_{k+1}^{(n)}, S_n) - \mathcal{E}_{k-1}^{(n+1)} \cdot c(\mathcal{E}_{k-1}^{(n+1)}, S_n) \right] = 0$$

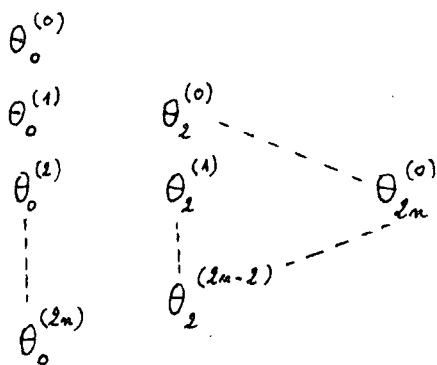
avec $\delta(\mathcal{E}_{-1}^{(n)}, S_n) = 0$, $\delta(\mathcal{E}_0^{(n)}, S_n) = \delta S_n$

$$c(\varepsilon_{-1}^{(n)}, S_n) = 0, \quad c(\varepsilon_0^{(n)}, S_n) = 1$$

Wynn a également montré que l' ε -algorithme possédait un effet de lissage (67). Si une même erreur δ affecte un certain nombre de quantités initiales S_n (mais pas toutes) alors cette erreur se propage en triangle mais diminue en passant d'une colonne paire à la colonne paire suivante. Cet effet de lissage est dû au pseudo-polynôme en δ_{in} qui apparaît dans la fonction $\Psi(n, a)$ correspondant à l' ε -algorithme (voir théorème II-2). Si toutes les quantités S_n ont la même erreur δ cette erreur est transmise à tous les stades ultérieurs du calcul; l' ε -algorithme ne peut en effet savoir si cette quantité δ fait ou non partie des S_n .

5-2 Application des algorithmes

Il y a deux façons essentielles d'utiliser des algorithmes de la forme (V-1). La première est une utilisation a posteriori. Si nous avons à notre disposition $2n+1$ quantités initiales $\theta_0^{(n)}$ nous pouvons leur appliquer l'algorithme et construire le tableau triangulaire correspondant :



La précision obtenue sera estimée, nous l'avons vu, par $\theta_{2k}^{(n)} - S \sim D_{2k-1}^{(n)}$.

si $\lim_{n \rightarrow \infty} \theta_0^{(n)} = S$.

Cette estimation de la précision peut précisément être utilisée pour une programmation en parallèle des algorithmes. Cette programmation en parallèle consiste à calculer successivement les quantités initiales $\theta_0^{(n)}$ et à appliquer l'algorithme au fur et à mesure. On arrêtera le processus quand la précision estimée sera suffisante. On peut également mettre en oeuvre des applications répétées de ces algorithmes. Il est en effet possible de recommencer l'application de l'algorithme en prenant, dans le tableau triangulaire obtenu, certaines quantités comme conditions initiales pour une nouvelle application. Posons $\theta_k^{(n)} \equiv \theta_k^{(n)}$.

1°) Application répétée associée

On obtient une nouvelle suite $\theta_k^{(n)}$ en prenant

$${}_s \theta_{-1}^{(n)} = 0 \quad {}_s \theta_0^{(n)} = {}_{s-1} \theta_{2n}^{(0)} \quad n = 0, 1, \dots$$

2°) Application répétée correspondante

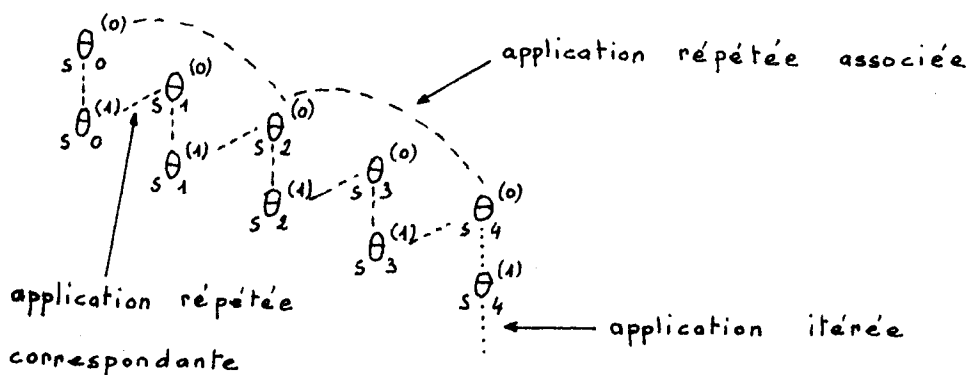
on prend ${}_s \theta_{-1}^{(n)} = 0$, ${}_s \theta_0^{(2n)} = {}_{s-1} \theta_n^{(0)}$ et

$${}_s \theta_0^{(2n+1)} = {}_{s-1} \theta_n^{(1)} \quad \text{pour } n = 0, 1, \dots$$

3°) Application itérée

On prend ${}_s \theta_{-1}^{(n)} = 0$, ${}_s \theta_0^{(n)} = {}_{s-1} \theta_{2k}^{(n)}$ $n = 0, 1, \dots$ pour $k > 0$

quelconque on a :



Les applications répétées associée et correspondante sont liées aux fractions continues associée et correspondante (58). Sur des exemples on s'aperçoit que l'application associée donne de meilleurs résultats mais que l'application correspondante est plus simple à mettre en oeuvre.

Supposons que nous disposions de $2N+1$ valeurs initiales $\theta_0^{(n)}$. A partir de ces valeurs on peut soit calculer $\theta_{2N}^{(n)}$ soit par applications itérées aboutir à une seule valeur finale. On montre que dans tous les cas on est conduit à $N(2N+1)$ calculs des quantités $D_k^{(n)}$ ce qui nous fournit le nombre total d'opérations à effectuer soit $2N(2N+1)$ additions-soustractions et $N(2N+1)$ divisions pour l' ϵ -algorithme et $3N(2N+1)$ additions - soustractions et $N(2N+1)$ divisions pour le ξ -algorithme simplifié.

La programmation proprement dite des algorithmes a été effectuée en employant la technique du losange de Wynn (60,61). Seule l'estimation de la précision a été modifiée car elle nous semblait inadéquate.

On retrouvera en annexe les listing de trois programmes FORTRAN. Le premier effectue la programmation en parallèle de l' ϵ -algorithme, le second celle du ξ -algorithme. Dans ces deux programmes le branchement vers les règles particulières est commandé par la variable NC suivant la méthode préconisée par Wynn pour le contrôle de la perte de précision due à la cancellation. Sur les nombreux exemples qui ont été traité on s'aperçoit que lors d'une sortie par la première étiquette la valeur de RES obtenue à NC chiffres significatifs exacts au minimum. Afin d'éviter certains ennuis de programmation quand les suites tendent vers zéro par valeurs très faibles, nous avons systématiquement utilisé le fait que $\bar{\theta}_{2k}^{(n)} = a \cdot \theta_{2k}^{(n)}$ si $\bar{\theta}_0^{(n)} = a \cdot \theta_0^{(n)}$ et toujours choisi a de sorte que $\bar{\theta}_0^{(0)} = 1$. Il existe également des versions spéciales de ces sous programmes pour des suites de nombres complexes.

Le troisième programme de l'annexe est relatif à la première forme confluente de l' ϵ -algorithme. Il permet, à partir d'une fonction FORTRAN, à écrire par l'utilisateur et qui calcule la valeur de la fonction et de ses dérivées au point d'abscisse t , de calculer $\epsilon_{2k}(t)$ et R_k pour $k=1, \dots, KM$.

C H A P I T R E VI

APPLICATIONS ET RESULTATS NUMERIQUES

Ce chapitre est consacré aux applications et aux résultats numériques des formes confluentes et discrètes des algorithmes étudiés. On applique les formes discrètes aux limites de suites et aux sommes de séries. On démontre que l'application de l' \mathcal{E} -algorithme aux méthodes itératives du premier ordre pour résoudre un système linéaire conduit à la solution exacte même si la méthode est divergente. On étudie également la résolution des équations et l'on donne un nouvel algorithme de résolution des systèmes d'équations non linéaires. Cette méthode est d'ordre deux, et nécessite ni calculs de dérivées ni inversions de matrices et converge sous des hypothèses assez peu restrictives. On traite ensuite de l'interpolation inverse puis on compare les algorithmes à la méthode de Romberg pour le calcul des intégrales définies et la dérivation numérique. On propose un nouveau procédé basé sur le \mathcal{E}^2 -algorithme et qui donne de meilleurs résultats que la méthode de Romberg. Les formes confluentes sont testées sur le calcul de la limite d'une fonction, la résolution des équations et le calcul des intégrales impropres. Enfin on donne une seconde forme discrète de l' \mathcal{E} -algorithme.

6-1 Applications des formes discrètes

Les algorithmes discrets étudiés sont les suivants :

1°) ε -algorithme
$$\varepsilon_{k+1}^{(n)} = \varepsilon_{k-1}^{(n+1)} + \frac{1}{\varepsilon_k^{(n+1)} - \varepsilon_k^{(n)}}$$

2°) \wp -algorithme simplifié
$$\wp_{k+1}^{(n)} = \wp_{k-1}^{(n+1)} + \frac{k+1}{\wp_k^{(n+1)} - \wp_k^{(n)}}$$

3°) forme étendue du \wp -algorithme
$$\wp_{k+1}^{(n)} = \wp_{k-1}^{(n+1)} + \frac{x_{n+k+1} - x_n}{\wp_k^{(n+1)} - \wp_k^{(n)}}$$

4°) \wp^r -algorithme
$$\wp_{k+1}^{(n)} = \wp_{k-1}^{(n+1)} + \frac{x_{n+k+1}^r - x_n^r}{\wp_k^{(n+1)} - \wp_k^{(n)}}$$

Les résultats numériques seront présentés de deux façons différentes. La première correspond plus précisément à l'application a posteriori des algorithmes

Elle consiste à donner le tableau triangulaire obtenu :

$$\begin{array}{cccc} \theta_0^{(0)} & & & \\ \theta_0^{(1)} & \theta_2^{(0)} & & \\ \theta_0^{(2)} & \theta_2^{(1)} & \theta_4^{(0)} & \\ \theta_0^{(3)} & \theta_2^{(2)} & \vdots & \\ \theta_0^{(4)} & \vdots & \vdots & \end{array}$$

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 maximaux que l'on peut atteindre dans le tableau triangulaire à partir des valeurs initiales $\theta_0^{(n)}$:

$$\theta_0^{(0)} \quad \theta_0^{(0)}$$

$$\theta_0^{(1)} \quad \theta_0^{(1)}$$

$$\theta_0^{(2)} \quad \theta_2^{(0)}$$

$$\theta_0^{(3)} \quad \theta_2^{(1)}$$

$$\theta_0^{(2n)} \quad \theta_{2n}^{(0)}$$

$$\theta_0^{(2n+1)} \quad \theta_{2n}^{(1)}$$

Les résultats seront systématiquement présentés sous l'une de ces 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.

6.1.1. Limites de suites et sommes de séries

Nous avons appliqué l' \mathcal{E} -algorithme à la suite totalement monotone

$$S_n = 1 + \frac{1}{n+1} \quad n \geq 0. \text{ On vérifie facilement que } \mathcal{E}_{2k}^{(n)} = \frac{S_{n+k} + k}{k+1} \text{ et que}$$

par conséquent, la convergence n'est qu'améliorée et non pas accélérée :

2,0000000					
1,5000000	1,2500000				
1,3333333	1,1666667	1,1111109			
1,2500000	1,1249999	1,0333337	1,0624991		
1,2000000	1,1000001	1,0666663	1,0500028	1,0399799	
1,1666667	1,0833333	1,0555563	1,0416545	1,0334257	
1,1428571	1,0714287	1,0476161	1,0357504		
1,1250000	1,0624998	1,0416716			
1,1111111	1,0555557				
1,1000000					

Considérons maintenant la suite totalement oscillante décalée $S_n = 1 + \frac{(-1)^n}{n+1}$

pour $n \geq 0$. La convergence est alors accélérée avec l' ϵ -algorithme :

2,0000000					
0,5000000	1,0357143				
1,3333333	0,99019608	1,0011455			
0,7500000	1,0040323	0,99974041	1,0000349		
1,2000000	0,99795918	1,0000832	0,99999271	1,0000010	
0,8333334	1,0011737	0,99996696	1,0000020	0,99999981	
1,1428571	0,99926363	1,0000152	0,99999931		
0,8750000	1,0004921	0,99999225			
1,1111111	0,99965493				
0,9000001					

Quand la sommation des séries un des exemples les plus spectaculaires est la série qui donne $\log (1+x)$:

$$\text{Log}(1+x) = x - \frac{x^2}{2} + \frac{x^3}{3} - \frac{x^4}{4} + \dots + (-1)^{n+1} \frac{x^n}{n} + \dots$$

on sait que cette série est convergente pour $x \in] -1, +1]$

Pour $x=1$ on doit trouver $\text{Log } 2 = 0,6931471805599453$

SUITE INITIALE	ALGORITHME	précision relative estimée
.1000000000000000+001	.1000000000000000+001	.10+101
.5000000000000000+000	.5000000000000000+000	.10+101
.8333333333333333+000	.7000000000000000+000	.10+001
.5833333333333333+000	.6904761904761905+000	.60+000
.7833333333333333+000	.6933333333333333+000	.22+001
.6166666666666667+000	.6930894303943089+000	.28-001
.7595238095238095+000	.6931524547803618+000	.29-001
.6345238095238095+000	.6931457431457431+000	.11-002
.7456349206349206+000	.6931473323543808+000	.87-003
.6456349206349206+000	.6931471424877166+000	.55-004
.7365440115440115+000	.6931471849621316+000	.36-004
.6532106782106782+000	.6931471795177768+000	.31-005
.7301337551337551+000	.6931471806881643+000	.18-005
.6587051837051837+000	.6931471805308536+000	.19-006
.7253718503718504+000	.6931471805636898+000	.97-007
.6628718503718504+000	.6931471805591229+000	.12-007
.7216953797836151+000	.6931471805600549+000	.57-008
.6661398242280595+000	.6931471805599219+000	.78-009
.7187714031754279+000	.6931471805599485+000	.35-009
.6687714031754279+000	.6931471805599446+000	.52-010
.7163904507944756+000	.6931471805599454+000	.22-010
.6709359053399301+000	.6931471805599453+000	.35-011

La notation .674+001 signifie 0.674×10^1

Un fait assez remarquable est que l' ϵ -algorithme converge toujours aussi rapidement vers $\text{Log}(1+x)$ et ceci même si la série est divergente. Pour $x=2$, nous avons $\text{Log } 3 = 1,098612288668110$ et :

.2000000000000000+001	.2000000000000000+001	.10+101
.0000000000000000+000	.0000000000000000+000	.10+101
.2666666666666667+001	.1142857142857143+001	.27+000
-.1333333333333333+001	.1066666666666667+001	.83+000
.5066666666666667+001	.1101449275362319+001	.10+002
-.5600000000000000+001	.1097046413502110+001	.11+000
.1268571428571429+002	.1098805646036916+001	.18+000
-.1931428571428571+002	.1098521046643914+001	.66-002

.3757460317460317+002	.1098625759028444+001	.75-002
-.6482539682539683+002	.1098606544543604+001	.49-003
.1213564213564214+003	.1098613236862854+001	.45-003
-.2199769119769120+003	.1098611911891433+001	.41-004
.4101769341769342+003	.1098612355798595+001	.33-004
-.7601087801087801+003	.1093612263343932+001	.38-005
.1424424553224553+004	.1098612293437743+001	.27-005
-.2671575446775447+004	.1098612286939119+001	.37-006
.5038542200283377+004	.1098612289007790+001	.23-006
-.9525013355272179+004	.1098612288548797+001	.37-007
.1806909190788572+005	.1098612288692340+001	.22-007
-.3435970809211428+005	.1098612288659814+001	.37-008
.6550467286026667+005	.1098612288669840+001	.21-008
-.1251455089579151+006	.1098612288667530+001	.38-009
.2395765779986066+006	.1098612288668233+001	.21-009

Dans ces deux cas les S_n sont évidemment les sommes partielles successives des séries. Sur cet exemple on peut également montrer comment la forme étendue du \mathcal{P} -algorithme améliore les résultats par rapport à la forme simplifiée lorsque la dépendance en n n'est pas une simple numérotation. Prenons en effet

$$S_n = \frac{1}{\log 2} \sum_{i=0}^{2^n} \frac{(-1)^i}{i+1} ; \text{ on a } \lim_{n \rightarrow \infty} S_n = 1.$$

Avec la forme simplifiée du \mathcal{P} -algorithme on obtient :

1,4426950			
0,72134753	0,92744680		
0,84157211	1,2245098	1,0352688	
0,91542436	1,0987293	0,92552137	0,96337200
0,95632194	1,0468760	0,95878550	
0,97780995	1,0229429		
0,98881698			

Avec la forme étendue du \mathcal{P} -algorithme on obtient, en prenant $x_n = 2^n$:

1,4426950
 0,72134753 0,88781232
 0,84157211 1,0014584 1,0001780
 0,91542436 1,0002609 0,99999791 0,99999983
 0,95632194 1,0000385 0,99999991
 0,97780995 1,0000051
 0,98881698

Signalons qu'il faudrait 10^{16} termes de la série pour obtenir le même résultat pour Log 2.

6.1.2. Méthodes itératives en analyse numérique linéaire

L'analyse numérique linéaire offre des procédés itératifs qui peuvent être facilement accélérés à l'aide de l' ϵ -algorithme. Ces procédés sont des procédés linéaires du premier ordre. Nous allons spécialement étudier deux d'entre eux :

Une méthode itérative de résolution de système et la méthode de la puissance itérée (ou de Krylov) pour le calcul du rayon spectral d'une matrice.

6.1.2.1. Résolution de système linéaire

Soit à résoudre le système $n \times n$: $Ax = b$ à l'aide d'une méthode de la forme $x^{(k+1)} = Bx^{(k)} + c$. La solution x du système doit évidemment vérifier $x = Bx + c$, d'où en soustrayant :

$$x^{(k)} = x + b^k (x^{(0)} - x)$$

On sait qu'il existe une matrice $n \times n$ non singulière Y qui réduit B à sa forme normale de Jordan J (24) :

$$J = Y \cdot B \cdot Y^{-1} \quad \text{avec } J = \begin{pmatrix} J_1 & & & \\ & J_2 & & \\ & & \dots & \\ & & & J_k \end{pmatrix}$$

où les J_i sont des matrices $n_i \times n_i$ (où n_i est la multiplicité de la valeur propre λ_i) :

$$J_i = \begin{pmatrix} \lambda_i & 1 & & & 0 \\ & \lambda_i & 1 & & \\ & & \lambda_i & \ddots & \\ & & & \ddots & 1 \\ 0 & & & & \lambda_i \end{pmatrix}$$

Soient y_i les vecteurs colonnes de Y . $x^{(0)}$ - x peut se développer suivant cette base. On aura $B^k Y = Y J^k$ avec J^k diagonale par blocs. La $j^{\text{ième}}$ composante de $B^k y_i$ sera de la forme $A_i(k) \lambda_i^k$ où $A_i(k)$ est un polynôme en k de degré $n_i - 1$ si λ_i est réelle, de la forme $\begin{bmatrix} B_i(k) \cos b_i k + c_i(k) \sin b_i k \end{bmatrix} \cdot \lambda_i^k$ si λ_i est complexe ou enfin de la forme d'un pseudo-polynôme en δ de degré $n_i - 1$ si λ_i est nulle. En regroupant les valeurs propres égales on voit que :

$$x_i^{(k)} = x + \sum_{j=1}^p A_j(k) e^{a_j k} + \sum_{j=p+1}^q \left[B_j(k) \cos b_j k + C_j(k) \sin b_j k \right] e^{a_j k} + \sum_{j=0}^m c_j \delta^{j i} \quad i = 1, \dots, n$$

avec $\sum_{j=1}^p (n_i - 1) + 2 \sum_{j=p+1}^q (n_i - 1) + m + 1 = n$ puisque

la somme de toutes les multiplicités des valeurs propres est égale à n . On voit donc que l'application de l' \mathcal{E} -algorithme aux suites des itérées successifs de chaque composante est telle que si $\mathcal{E}_0^{(k)} = x_i^{(k)}$ $k=0, \dots, 2n$ alors on obtient $\mathcal{E}_{2n}^{(0)} = x_i \quad \forall i = 1, \dots, n$. On est donc capable avec l' \mathcal{E} -algorithme d'obtenir la solution exacte d'un système linéaire $n \times n$ à partir de $2n+1$ itérés successifs fournis par une méthode itérative linéaire d'ordre un et ceci même si la méthode est divergente c'est à dire même si $\rho(B) > 1$.

Soit à résoudre
$$\begin{pmatrix} 1 & 2 & -2 \\ 2 & 2 & 2 \\ 2 & 2 & 1 \end{pmatrix} x = \begin{pmatrix} -3 \\ 0 \\ -1 \end{pmatrix}$$
 par la méthode de Jacobi.

La solution de ce système est $x_1 = -1$, $x_2 = 0$ et $x_3 = 1$.

Si on applique l' ε -algorithme sur les itérations de la première composante on obtient :

0			
0,5	1,0000000000000000		
1,25	-2,0000000000000000	-0,2000000000000000	
1,625	-1,35714285714286	-1,0000000000000000	-1,0000000000000000
0,6875	0,2500000000000000	-1,0000000000000000	
0,21875	-0,553571428571429		
1,390625			

Les résultats pour les autres composantes sont analogues à ceux-ci.

L' ε -algorithme apparait donc ainsi comme une méthode capable d'accélérer les méthodes itératives linéaires du premier ordre pour la résolution des systèmes linéaires.

6.1.2.2. Calcul du rayon spectral d'une matrice

Soit à calculer le rayon spectral de la matrice A par la méthode de

la puissance itérée :

Soit $x^{(0)}$ un vecteur donné tel que $\|x^{(0)}\| = 1$. On forme la suite :

$$\left. \begin{aligned} z^{(k+1)} &= A x^{(k)} \\ x^{(k+1)} &= z^{(k+1)} / \|z^{(k+1)}\| \end{aligned} \right\} k=0,1,\dots$$

et on a $\lim_{k \rightarrow \infty} \|z^{(k)}\| = \rho(A)$

Appliquons cette méthode à la matrice triple diagonale

$$A = \begin{pmatrix} 3 & -1 & & & \\ -1 & 3 & -1 & & 0 \\ & -1 & 3 & -1 & \\ & & 0 & -1 & 3 & -1 \\ & & & & -1 & 3 \end{pmatrix} \quad \rho(A) = 4,732050807568877$$

A partir de 23 itérés l' ϵ -algorithme donne $\rho(A)$ avec 15 chiffres exacts alors que la meilleure des valeurs initiales n'en avait que 3. Donnons la dernière partie du tableau ϵ (à partir de l'itéré 17)

4,72329114099681
 4,72648711474843 4,73206876039121
 4,72851941897166 4,73205802782436 4,73205079106963
 4,72981032579747 4,73205371078885 4,73205080334393 4,73205080757428
 4,73062972644351 4,73205197477057 4,73205080648659
 4,73114660766131 4,73205127678296
 4,73147936090888

Et on a $\epsilon_{22}^{(0)} = 4,7320508756887$

Si la matrice à deux valeurs propres voisines la suite initiale oscille lentement entre ces deux valeurs et l' ϵ -algorithme converge soit vers l'une soit vers l'autre suivant le vecteur initial $x^{(0)}$.

Nous avons pu accélérer également le calcul de la plus grande valeur propre de la matrice de Jacobi associée à la méthode de sur-relaxation.

$$\epsilon_o^{(n)} = \frac{\|x^{(n+2)} - x^{(n+1)}\|_2}{\|x^{(n+1)} - x^{(n)}\|_2}$$

$\| \cdot \|_2$ désignant la norme euclidienne

Pour la matrice triple diagonale 200x200

$$A = \begin{pmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & 0 \\ & -1 & 2 & -1 & \\ & & 0 & \dots & -1 \\ & & & \dots & -1 & 2 \end{pmatrix} \quad \rho^2(B) = 0,999755728800233$$

nous avons obtenu avec l' ϵ -algorithme :

Suite initiale	algorithme
0,372677996249964949	0,372677996249964949
0,667707520800337648	0,667707520800337648
0,775009909605131679	0,836342559237732058
0,829614143306923340	0,886193397975935624
0,862709492911615405	0,925597103073743300
0,884941725686843855	0,942245069279620419
0,900920703529149200	0,957609212256559969
0,912967766208480546	0,965006859905541171

Bien que cela n'apparaisse pas le gain obtenu est très appréciable pour la suite de l'application de la méthode de sur-relaxation.

6.1.3. Résolution des équations

L' ϵ -algorithme permet d'accélérer des méthodes itératives de résolution des équations. Si la méthode est d'ordre deux nous avons vu que l' ϵ -algorithme est incapable d'accélérer la suite des itérés. C'est ce qui se passe pour la résolution de $f(x) = x - e^{-x}$ par la méthode de Newton :

Suite initiale	algorithme	précision
.0000000000000000+000	.0000000000000000+000	.10+101
.5000000000000000+000	.5000000000000000+000	.10+101
.5663110031972182+000	.7564499487951879+000	.10+001
.5671431650348622+000	.5671537408683628+000	.15-002
.5671432904097810+000	.5671431471513406+000	.32-007
.5671432904097839+000	.5671432904097810+000	.33-010
.5671432904097839+000	.5671432904098739+000	.00+000

En utilisant dans ce cas la méthode d'Overholt exposée au paragraphe 2.6. on obtient une accélération notable de la convergence :

$$x_2^{(1)} = 0,576449948795187889$$

$$x_3^{(1)} = 0,567152276606922667 \quad \text{4 chiffres exacts}$$

$$x_4^{(1)} = 0,567143290428435828 \quad \text{10 chiffres exacts au lieu de 6}$$

$$x_5^{(1)} = 0,5671432900409783871 \quad \text{17 chiffres exacts au lieu de 14}$$

Même dans des cas de convergence linéaire la méthode d'Overholt apporte une accélération par rapport à l' ϵ -algorithme. Si nous résolvons $x = e^{-x}$ par le procédé du premier ordre $x_{n+1} = e^{-x_n}$ nous obtenons $\frac{dy}{dx} = 0$

suite initiale	!	ϵ -algorithme	!	chiffres ex.	!	méth.d'Overholt	!	chif.ex.
1,000000	!	1,000000	!	0	!	1,000000	!	0
0,367879		0,612699		0		0,612699		0
0,692200		0,582226		1		0,571338		1
0,500473		0,564857		2		0,566958		2
0,606243		0,567526		3		0,567134		4
0,545395		0,567170		4		0,567143		6
0,579612		0,567146		5				8
0,560115		0,567143		6				9

Si le procédé est du premier ordre on voit que l'accélération obtenue à l'aide de l' ϵ -algorithme est notable.

Appliquons la méthode de Newton à la recherche de la racine double $x = -3$ du polynôme $x^6 + 6x^5 + 6x^4 - 18x^3 - 31x^2 - 24x - 36$ en partant de $x^{(0)} = -3,5$. Deux phénomènes se produisent : une accélération de la convergence car la méthode de Newton est d'ordre un pour une racine multiple et une amélioration de la précision du résultat car à partir d'un certain rang les itérés de la méthode de Newton se mettent à osciller et il est alors impossible d'améliorer la précision :

suite initiale	algorithme
-.3500000000000000+001	-.3500000000000000+001
-.3314015312632922+001	-.3314015312632922+001
-.3186974034742264+001	-.2913160788588798+001
-.3105694680781266+001	-.2961331576850768+001
-.3057165552657127+001	-.3021542742532907+001
-.3029932985613312+001	-.3004720560384673+001
-.3015351668562098+001	-.2998019945539767+001
-.3007779399885537+001	-.2999797914487431+001
-.3003916605819156+001	-.3000052617516176+001
-.3001965163736187+001	-.3000002437371561+001
-.3000984314375852+001	-.2999999638429583+001
-.3000492592510335+001	-.2999999992351310+001
-.3000246405362310+001	-.3000000000619666+001
-.3000123229992691+001	-.300000000006568+001
-.3000061621828629+001	-.299999999999613+001
-.3000030812623127+001	-.3000000000001545+001
-.3000015406738995+001	-.3000000000000763+001
-.3000007703475447+001	-.3000000000001797+001
-.3000003851764188+001	-.299999999999818+001

Nous avons également appliqué l' ϵ -algorithme aux suites d'approximations de la somme et du produit de deux racines d'un polynôme obtenues par la méthode de Bairstow. Quand l'une de ces deux racines est multiple la méthode est d'ordre 1 et peut donc être accélérée par l' ϵ -algorithme.

C'est ce qui se passe pour le polynôme $x(x - 8,01)^5$ où les deux premières racines trouvées en partant de $s = p = 0$ sont $x = 0$ et $x = 8,01$. On obtient alors pour les itérés de la somme de ces racines (en n'appliquant l' ϵ -algorithme qu'à partir du sixième itéré pour éviter les oscillations) :

suite initiale	algorithme
.591022656000+001	.591022656000+001
.633018124800+001	.633018124800+001
.666614499840+001	.801000000000+001
.693491599872+001	.801000000000+001
.714993279898+001	.801000000000+001
.732194623918+001	.801000000000+001

Signalons toutefois que l'accélération de la convergence n'est pas toujours aussi spectaculaire. Quand aucune des racines n'est multiple l'application de l' ϵ -algorithme n'apporte aucune amélioration puisque la méthode de Bairstow est d'ordre deux. Prenons, par exemple, le polynôme qui a pour racines 5, -5, $-6+i$, $4 \pm 3i$ et $3 \pm 4i$ on obtient pour la somme des racines $-6 + i$ et $-6 - i$:

suite initiale	algorithme
-.137315960545+002	-.137315960545+002
-.126242739239+002	-.126242739239+002
-.121121190121+002	-.116713980764+002
-.120043833285+002	-.119756829683+002
-.120000069798+002	-.120081384925+002
-.120000000000+002	-.120000086798+002
-.120000000000+002	-.119999995673+002
-.120000000000+002	-.119999995673+002

Quand des racines sont voisines à moins de 10^{-7} près l' ϵ -algorithme accélère la convergence. L' ϵ -algorithme apparait ainsi également comme un procédé permettant de savoir si une racine est multiple ou non. Il suffit pour cela de regarder si la convergence est accélérée ou non (le contrôle s'effectue grâce à la précision relative estimée).

Il existe une autre possibilité d'utilisation de l' ϵ -algorithme pour résoudre les équations et les systèmes d'équations non linéaires. C'est ce que je vais maintenant exposer (8,14).

Soit à résoudre $f(x)=0$ où f est une application de \mathcal{C}^P dans lui-même et soit s une solution. Nous allons étudier une méthode itérative d'ordre 2 pour le calcul de S . On rappelle qu'une suite $\{U_k\}$ qui converge vers u est d'ordre r si :

$$\lim_{k \rightarrow \infty} \frac{\|u_{k+1} - u\|}{\|u_k - u\|^r} = c \neq 0.$$

Soit F une application non forcément convergente de \mathcal{C}^P dans lui-même telle

que $s = F(s)$. On propose l'algorithme suivant :

Algorithme

$$\begin{array}{ll} & x_0 \text{ donné} \\ n^{\text{ième}} \text{ pas} & u_0 = x_0 \\ & u_k = F(u_{k-1}) \quad k = 1, \dots, 2p \end{array}$$

Calcul de $\mathcal{E}_{2p}^{(0)}$ à l'aide de l' \mathcal{E} -algorithme vectoriel :

$$\mathcal{E}_{-1}^{(k)} = 0 \quad k = 0, \dots, 2p$$

$$\mathcal{E}_0^{(k)} = u_k \quad k = 0, \dots, 2p$$

$$\mathcal{E}_{k+1}^{(n)} = \mathcal{E}_{k-1}^{(n+1)} + \frac{\overline{\Delta \mathcal{E}_k^{(n)}}}{(\Delta \mathcal{E}_k^{(n)}, \Delta \mathcal{E}_k^{(n)})} \rightarrow \left[\begin{array}{l} k = 0, \dots, 2p-1 \\ n = 0, \dots, 2p-k-1 \end{array} \right.$$

$$x_{n+1} = \mathcal{E}_{2p}^{(0)}$$

avec $\Delta \mathcal{E}_k^{(n)} = \mathcal{E}_k^{(n+1)} - \mathcal{E}_k^{(n)}$, $\overline{\Delta \mathcal{E}_k^{(n)}}$ complexe conjugué de $\Delta \mathcal{E}_k^{(n)}$ et

$$(u, v) = \sum_{i=1}^p u_i \bar{v}_i \text{ produit scalaire dans } \mathbb{C}^p.$$

Démontrons maintenant le résultat fondamental suivant :

Théorème VI-1:

Soit à chercher $s \in \mathbb{C}^p$ tel que $f(s) = 0$ et soit F une application de \mathbb{C}^p dans lui même telle que :

$$S = F(s)$$

F différentiable dans un voisinage de s

$I - J$ inversible (J jacobien de F calculé en $x=s$)

alors il existe un voisinage de s tel, que quelque soit x_0 appartenant à ce

voisinage l'algorithme précédent converge vers s et soit d'ordre 2.

démonstration :

Si F est différentiable au sens de Fréchet on a :

$$u_{k+1} - s = J(u_k - s) + \varphi(u_k - s) \cdot \|u_k - s\| \quad \forall k \quad (\text{VI-1})$$

où J est le jacobien de F calculé en $x = s$ et où

$$\lim_{h \rightarrow 0} \frac{\varphi(h)}{\|h\|} = 0$$

supposons que les $2p+1$ vecteurs $u_0 = x_n, u_1 = F(u_0), \dots, u_{2p} = F(u_{2p-1})$ soient générés par une relation de la forme :

$$u_{k+1} = B_n u_k + b_n \quad (\text{VI-2})$$

où B_n est une matrice qui dépend de x_n . Appelons x_{n+1} le point fixe de $F_n(x) =$

$$B_n x + b_n$$

on sait (72) que les vecteurs u_k produits par une relation de la forme (VI-2) vérifient l'équation aux différences :

$$\sum_{k=0}^p c_k u_{k+p} = x_{n+1} \sum_{k=0}^p c_k \quad \forall p \quad (\text{VI-3})$$

où les c_k sont les coefficients de l'équation caractéristique de la matrice B_n .

On sait d'autre part (13) que l' \mathcal{E} -algorithme revient à chercher les coefficients

c_k de cette équation aux différences puis à calculer x_{n+1} . Donc l'utilisation

de l' \mathcal{E} -algorithme revient à supposer que u_0, \dots, u_{2p} sont générés par une

relation du type (VI-2) :

De (VI-3) on déduit :

$$\sum_{k=0}^p c_k (u_k - s) = (x_{n+1} - s) \sum_{k=0}^p c_k \quad (\text{VI-4})$$

et de (VI-2) $x_{n+1} - u_{k+1} = B_n (x_{n+1} - u_k)$

Effectuons la différence avec (VI-1), multiplions par c_k et sommons :

$$\begin{aligned} (I-B_n) (x_{n+1}-s) \sum_{k=0}^p c_k &= (J-B_n) \sum_{k=0}^p c_k (u_k-s) \\ &+ \sum_{k=0}^p c_k \cdot \varphi(u_k-s) \cdot \|u_k-s\| \end{aligned}$$

Utilisons (VI-4) :

$$(I-B_n) (x_{n+1}-s) \sum_{k=0}^p c_k = (J-B_n) (x_{n+1}-s) \sum_{k=0}^p c_k + \sum_{k=0}^p c_k \cdot O(\|u_k-s\|^2)$$

d'où :

$$(I-J) (x_{n+1}-s) \sum_{k=0}^p c_k = \sum_{k=0}^p c_k \cdot O(\|u_k-s\|^2)$$

Or d'après (VI-1) $u_{k+1}-s = O(u_k-s)$ et par conséquent $u_k-s = O(x_n-s) \forall k$ puisque $u_0 = x_n$. D'où finalement $O(\|u_k-s\|^2) = O(\|x_n-s\|^2) \forall k$ et :

$$(I-J) (x_{n+1}-s) \sum_{k=0}^p c_k = O(\|x_n-s\|^2)$$

ce qui démontre que, si l'on peut appliquer notre algorithme, alors le procédé est d'ordre 2.

On sait (13) qu'une condition nécessaire et suffisante pour que l'on puisse appliquer l' ε -algorithme est que $\sum_{k=0}^p c_k \neq 0$. Cela signifie que B_n ne doit pas admettre 1 comme valeur propre et donc que $I - B_n$ ne doit pas avoir de valeur propre nulle. $I-B_n$ doit être inversible. Etudions maintenant la condition pour que $(I-B_n)^{-1}$ existe et pour que l'itération converge à partir d'un certain x_0 .

Montrons d'abord que B_n tend vers J quand x_n tend vers s . Soit ϕ l'application qui à x_n fait correspondre x_{n+1} par notre algorithme. Exprimons la quantité $\phi(s+h) - \phi(s)$. Soit B_1 la matrice correspondant à $u_0 = s+h$.

En supposant la régularité de $I - B_1$ et en remarquant que $S = \phi(s)$, on peut écrire :

$$\begin{aligned} \phi(s+h) - \phi(s) &= (I-B_1)^{-1} [F(s+h) - B_1(s+h)] - s \\ &= (I-B_1)^{-1} [s+Jh + \varphi(h) \cdot \|h\| - B_1(s+h)] - s \end{aligned}$$

puisque $s = F(s)$. D'où finalement :

$$\phi(s+h) - \phi(s) = (I-B_1)^{-1} (J - B_1) h + (I-B_1)^{-1} \cdot \varphi(h) \cdot \|h\|$$

avec $\lim_{h \rightarrow 0} \varphi(h) = 0$. Comme la méthode est d'ordre 2, la matrice $(I-B_1)^{-1}(J-B_1)$ doit tendre vers la matrice nulle lorsque h tend vers zéro, ce qui démontre que B_n tend vers J quand x_n tend vers s .

Supposons maintenant $I - J$ inversible. Puisque $I-B_n$ tend uniformément vers $I-J$ on sait $\exists \eta_1 > 0 : \forall x_n \in B(s, \eta_1)$ alors $(I-B_n)^{-1}$ existe. D'autre part on a :

$$\|x_{n+1} - s\| \leq \| (I-B_n)^{-1} \| \cdot (\| J-B_n \| + \| \varphi(x_n - s) \|) \cdot \| x_n - s \|$$

donc à cause de la convergence de B_n vers J , $\exists \eta_2 > 0 : \forall x_n \in B(s, \eta_2)$ alors :

$$\| (I-B_n)^{-1} \| \cdot (\| J-B_n \| + \| \varphi(x_n - s) \|) < 1$$

ce qui démontre le théorème VI-1.

Si l'on possède des informations supplémentaires sur la convergence de $u_{k+1} - s - J(u_k - s)$ vers zéro lorsque u_k tend vers s , alors on peut améliorer le résultat du théorème VI-1 par :

Théorème VI-2 :

Si pour $\lambda > 1$ on a $F(x) - s - J(x-s) = O(\|x-s\|^\lambda)$ alors la suite générée par notre algorithme est telle que

$$x_{n+1} - s = O(\|x_n - s\|^\lambda)$$

De même si $F(x) - s - J(x-s) = o(\|x-s\|^\lambda)$ alors :

$$x_{n+1} - s = o(\|x_n - s\|^\lambda)$$

La démonstration est immédiate en remplaçant, dans la démonstration du théorème VI-1, $\varphi(u_k - s) \cdot \|u_k - s\|$ par $O(\|u_k - s\|^\lambda)$ ou $o(\|u_k - s\|^\lambda)$. Ce résultat est une généralisation de celui donné par Ostrowski (44) dans le cas $p=1$.

Remarques :

1°) Si $p=1$ l'algorithme proposé se réduit à la méthode de Steffensen. Il en possède d'ailleurs les propriétés et l'on peut donc le considérer comme sa généralisation à plusieurs dimensions.

2°) Si F est linéaire, $\varphi(u_k - s) = 0 \quad \forall k$, $B_n = J$ d'où $x_{n+1} = s \quad \forall x_n$. On retrouve bien le fait que cet algorithme fournit la solution d'un système linéaire en une itération quelque soit le vecteur initial et ceci même si les itératives de base divergent (15).

3°) La construction d'une méthode d'ordre 2 à partir d'une itération de base du premier ordre revient à supprimer les termes d'ordre 1 dans (VI-1) à la limite. La méthode de Newton effectue cette suppression en approchant le jacobien en s par le jacobien en x_n . L'algorithme étudié ici revient à approcher (VI-1) par (VI-2). D'autre part, on sait (15) que les vecteurs produits par (VI-2) sont tels que :

$$u_k = x_{n+1} + \sum_{i=1}^m P_i(k) \lambda_i^k y_i$$

où les inconnues, qui sont les λ_i et les coefficients des polynômes $P_i(k)$, sont au nombre de $2p$. La détermination de x_{n+1} nécessite donc l'écriture de $2p$ équations (VI-2) et, par conséquent la connaissance de $2p+1$ vecteurs u_k ; d'où le résultat :

Il est impossible, sans calculer de dérivées ni inverser de matrices, de construire une méthode d'ordre 2 telle que le passage de x_n à x_{n+1} nécessite moins

de $2p$ évaluations de F . En ce sens l'algorithme proposé est optimal .

Ulm a proposé (53) sous le nom d'extension de la méthode de Steffensen, un algorithme qui est également d'ordre 2, mais qui est loin d'être optimal car il nécessite beaucoup plus d'évaluation de F ainsi que des inversions de matrices.

4°) Si $J=0$ les itérations de base $u_{k+1} = F(u_k)$ sont d'ordre supérieur à 1.

L'algorithme est alors d'ordre supérieur à 2 mais les résultats qu'il donne sont moins bons que ceux obtenus à l'aide des itérations de base seules.

5°) Si l'algorithme converge alors il converge vers un point fixe de $x-F(x)=0$

En effet, supposons qu'il y ait convergence vers a alors :

$$a = \Phi(a) = (I-B)^{-1} \cdot [F(a) - Ba]$$

$$(I-B) a = F(a) - Ba \quad \text{d'où } a = F(a)$$

6°) A la place de l' ε -algorithme vectoriel on aurait pu utiliser l' ε -algorithme scalaire sur les suites des itérés de chacune des composantes car les deux algorithmes ont en commun la propriété (VI-3) qui est nécessaire pour les démonstrations. Cependant la programmation est simplifiée en utilisant l'algorithme vectoriel.

Donnons maintenant quelques exemples numériques :

Soit à résoudre le système :

$$\begin{aligned} x &= -\frac{y^4}{4} - \frac{3}{4} \\ y &= -0,405 e^{1-x^2} + 1,405 \end{aligned}$$

dont la solution unique est $x = 1$ et $y=1$. Les itérations de base convergent mais lentement car les valeurs propres de J valent $\pm 0,9$. En partant de $x_0 = y_0 = 0$ on obtient

Algorithme proposé

itération	x	y
1	-0,852296235398879384	0,870083624375792849
2	-0,969190257393206619	0,972676333650827051
3	-0,997913048775765137	0,997848573071205496
4	-0,999989311462832762	0,999984232445226238
5	-0,99999999717914572	0,999999999141560755
6	-0,99999999999999632	1,00000000000000250

Ce calcul a nécessité 24 évaluations de F

Méthode de Newton

Itération	x	Y
1	-0,750000000000000000	0,304095859474086683
2	-0,765876880623479763	0,792662514760822501
3	-0,921281538821868353	0,938407511361889756
4	-0,987935311564206382	0,991734013418315369
5	-0,999776520512802722	0,999874805260563668
6	-0,99999982677895456	1,000000006186878680

Avec les itérations de base on obtient :

Itération	x	y
20	-0,981427814583948488	0,984764782856446254
100	-0,999996212378911092	0,999996921974735428
200	-0,99999999899396917	0,99999999918244665
300	-0,999999999997325	0,9999999999997826

Soit à résoudre $x = \frac{y^2}{2} - \frac{3}{2}$

$$y = 0,605 e^{1-x^2} + 0,395$$

Dont la solution unique est $x = -1$ et $y = 1$. Les valeurs propres de J valent $\pm 1,1$ et les itérations de base divergent. En appliquant notre algorithme, à partir de $x_0 = y_0 = 0$, on obtient la solution avec 15 chiffres significatifs

exacts en 8 itérations.

Soit maintenant à calculer la racine réelle unique $x=1$ de $2x^3 - 15x^2 - 36x + 49 = 0$.

Si l'on part de $x_0 = 3 \pm \sqrt{12}$ la méthode de Newton oscille indéfiniment entre ces 2 valeurs. Avec les itérations de base $u_{k+1} = (2u_k^3 - 15u_k^2 + 49) / 36$ notre algorithme donne 18 chiffres exacts en 5 itérations avec $x_0 = 3 \pm \sqrt{12}$.

Soit enfin à résoudre $x = y, y = x^3$ qui possède la solution unique $x=y=0$.

Ce système vérifie les conditions du théorème VI-2 avec $\lambda=3$.

En partant de $x_0 = y = 0,5$ on obtient :

Itération	x	y
1	-0,233. 10 ⁻¹	-0,122. 10 ⁻¹
2	-0,325. 10 ⁻⁸	-0,273. 10 ⁻⁹
3	0,927. 10 ⁻²⁹	-0,927. 10 ⁻²⁹
4	0,796. 10 ⁻⁸⁷	-0,796. 10 ⁻⁸⁷
5	0,504. 10 ⁻²⁶¹	-0,504. 10 ⁻²⁶¹

6.1.4. Interpolation inverse

Soit $f(t)$ une fonction. L'interpolation inverse consiste à chercher la valeur de t telle que $f(t) = a$ donné au moyen d'une table $[t_n, f(t_n)]$ pour $n=0, \dots, k$.

On peut utiliser la forme étendue du ρ -algorithme pour résoudre ce problème

en prenant $\rho_0^{(n)} = t_n$ et $x_n = 1 / (f(t_n) - a)$

On aura $\lim_{t_n \rightarrow t} \frac{1}{f(t_n) - a} = \pm \infty$. Or les différences réciproques sont

complètement symétriques par rapport aux x_n . On peut donc considérer que

$\lim_{n \rightarrow \infty} x_n = +\infty$ même si les x_n ne sont pas rangés dans un ordre tel que cette

propriété soit vraie. Nous avons appliqué cette méthode à $f(t) = e^t$. En prenant

$a = e$ on doit trouver $t = 1$:

0			
0,3	0,977789169569660		
0,6	0,997706838803423	1,00003004984940	
0,9	1,00066500387999	0,999984709851802	1,00000076087482

1,2	0,999171228565960	1,00001741342502
1,5	1,00649439414436	
1,8		

en poursuivant le calcul on trouve $\varphi_{14}^{(0)} = 1,0000000161259$. On voit dans le tableau précédent qu'on peut également utiliser cette méthode pour l'extrapolation inverse. Pour $a = 12,18249396070347$ qui correspond à $t = 2,5$ on trouve :

3		
3,3	2,53441137162953	
3,6	2,59146301207838	2,50497180960558
3,9	2,68221249831015	
4,2		

De même pour $a = 3,49034295746181$ qui correspond à $t = 1,25$ on obtient :

0		
0,3	1,19108423852308	
0,6	1,23274511420969	1,24992970367604
0,9	1,24906495357945	
1,2		

Plus t et a sont en dehors de la table et moins les résultats obtenus sont précis. C'est d'ailleurs ce que l'on constate en comparant les extrapolations inverses correspondant à $t = 2,5$ et $t = 1,25$.

6.1.5. Comparaison avec l'extrapolation polynômiale

Nous allons comparer maintenant les résultats de l'extrapolation par fraction rationnelle fournis par le ξ^z -algorithme avec ceux de l'extrapolation polynômiale fournis par le procédé de Richardson.

L'extrapolation polynômiale est basée sur la formule d'interpolation de Lagrange. Soit $\{h_k\}$ une suite d'abscisses positives strictement décroissante et qui tend vers zéro quand k tend vers l'infini. Soit $f(h)$ une fonction connue aux points h_k pour $k=0, \dots, n$. Il existe un polynôme unique de degré n qui passe par ces $n+1$ points. Ce polynôme est donné par la formule de Lagrange :

$$P_n(h) = \sum_{k=0}^n A_k^n f(h_k)$$

avec

$$A_k^n = \prod_{\substack{j=0 \\ j \neq k}}^n \frac{h_j - h}{h_j - h_k}$$

Si nous extrapolons la valeur de ce polynôme en $h=0$ on obtient :

$$P_n(0) = \sum_{k=0}^n A_k^n f(h_k) \text{ avec } A_k^n = \prod_{\substack{j=0 \\ j \neq k}}^n \frac{1}{1 - \frac{h_k}{h_j}}$$

quand on passe du degré n au degré $n+1$ les coefficients sont liés par la relation :

$$A_k^{n+1} = \frac{h_{n+1}}{h_{n+1} - h_k} A_k^n \quad \text{pour } k=0, \dots, n$$

et

$$A_{n+1}^{n+1} = \prod_{j=0}^n \frac{1}{1 - \frac{h_{n+1}}{h_j}}$$

Supposons les pas en progression géométrique de raison $1/2$ et appelons T_k^n le résultat de l'extrapolation en zéro basé sur les abscisses

$$h_i = \frac{H}{2^i} \quad i = n, \dots, n+k$$

on a $T_0^n = f(h_n) \quad \forall n$ et le procédé de Richardson s'écrit :

$$T_{k+1}^{n+1} = \frac{2^k}{2^k - 1} T_k^{n+1} - \frac{1}{2^k - 1} T_k^n \quad (\text{VI-5})$$

Si nous voulons interpoler par un polynôme qui ne comporte que des puissances paires de h et si, ensuite, on extrapole pour $h = 0$ on retrouve le procédé de Romberg :

$$T_{k+1}^{n+1} = \frac{2^{2k}}{2^{2k} - 1} T_k^{n+1} - \frac{1}{2^{2k} - 1} T_k^n \quad (\text{VI-6})$$

Les quantités T_k^n peuvent être rangées dans un tableau à double entrée analogue au tableau \mathcal{E} et où l'indice inférieur désigne la colonne et l'indice supérieur la diagonale :

$$\begin{array}{cccc}
 & & & & T_0^0 \\
 & & & & T_1^0 \\
 & & & T_2^0 & T_3^0 \\
 & T_0^1 & T_1^1 & & \\
 & T_1^1 & & & \\
 & T_0^2 & T_1^2 & T_2^1 & \\
 & T_1^2 & & & \\
 & T_0^3 & & & \\
 & & & &
 \end{array}$$

Contrairement aux \mathcal{E} et \mathcal{P} -algorithmes toutes les colonnes ont une signification. On remarque cependant que le calcul de $T_{1,1}^0$ ne fait intervenir que T_0^0 et T_0^1 ; par contre celui de $T_{2,2}^0$ fait intervenir T_0^0 , T_0^1 et T_0^2 . On va donc pouvoir comparer $T_{2,2}^0$ avec $\mathcal{E}_2^{(0)}$ ou $\mathcal{P}_2^{(0)}$ et ainsi de suite en partant des mêmes valeurs initiales $\mathcal{E}_0^{(n)} = \mathcal{P}_0^{(n)} = T_0^n = S_n$ donné. Nous allons effectuer cette comparaison pour la quadrature et la dérivation numérique.

6.1.5.1. Application à la quadrature numérique

La méthode de Romberg est principalement employée pour améliorer le calcul numérique d'une intégrale définie. Soit $v(h)$ l'approximation de cette intégrale calculée par la méthode des trapèzes avec un pas h . $v(h)$ peut se mettre sous la forme :

$$v(h) = v(0) + \frac{h^2}{2!} v''(0) + \dots + \frac{h^{2n-2}}{(2n-2)!} v^{(2n-2)}(0) + h^{2n} S(h)$$

où $S(h)$ est une fonction bornée de h et où $v(0)$ représente la valeur exacte de cette intégrale. On montre que :

$$T_k^k - v(0) = \frac{h^{2k} v^{(2k)}(t)}{(2k)! (1.2.\dots.2^{k-1})^2}$$

Au lieu de prendre une approximation polynômiale pour $v(h)$ nous prendrons une approximation par fraction rationnelle. Si le numérateur et le dénominateur de

cette fraction rationnelle ne comportent que des puissances paires de h alors, en effectuant la division, nous obtiendrons un développement en puissances qui ne comportera que des puissances de h^2 et sera donc comparable au développement polynômial en h^2 correspondant à la méthode de Romberg. Cette extrapolation par fraction rationnelle s'effectuera à l'aide du ρ^2 -algorithme en prenant $x_n = 1/h_n$. D'où, si les pas sont en progression géométrique de raison $1/2$ (9) :

$$\rho_{k+1}^{(n)} = \rho_{k-1}^{(n+1)} + \frac{2^{2n} (2^{2k+2} - 1)}{\rho_k^{(n+1)} - \rho_k^{(n)}} \quad (\text{VI-7})$$

avec $\rho_{-1}^{(n)} = 0$ et $\rho_0^{(n)} = S_n =$ approximation de l'intégrale par la méthode des trapèzes avec un pas $H / 2^n$ $n = 0, 1, \dots$. Gragg (25) a montré que l'erreur d'une interpolation rationnelle est du même ordre de grandeur que celle de l'interpolation polynômiale pour des points d'interpolation identiques. (VI-7) fournira donc une précision comparable à (VI-6). La différence proviendra uniquement d'une meilleure représentation de $v(h)$ par un polynôme en h^2 pour h tendant vers zéro ou par une fraction rationnelle en $1/h^2$. Dans ce second cas, il est à prévoir que la dérivée 2^{nième} sera plus faible et, par conséquent l'erreur plus petite.

Nous allons comparer les résultats obtenus par la méthode de Romberg, l' \mathcal{E} -algorithme, la forme simplifiée du ρ -algorithme et le ρ^2 -algorithme (VI-7) appliqués aux estimations par la méthode des trapèzes avec $H = 1/3$ de

$$\int_0^1 \frac{dx}{x + 0,01} = 4,615120517$$

Nous obtenons les résultats suivants. (avec à chaque sommet de la table de haut en bas \mathcal{E} , ρ , ρ^2 et Romberg)

18,29

10,61 3,98
 3,66
 5,58
 5,72

7,06 4,32 4,84
 4,09 4,58
 4,89 4,65
 4,94 4,67

5,51 4,51 4,66 4,58
 4,38 4,6174 4,6176
 4,67 4,6199 4,61537
 4,68 4,6234 4,61570

4,91 4,59 4,62 4,613 4,61598
 4,53 4,6172 4,6173 4,615106
 4,62 4,6154 4,6151273 4,615120586
 4,62 4,6157 4,6151402 4,615120793

4,69 4,611 4,6156 4,61504
 4,59 4,6156 4,61509
 4,6157 4,6151293 4,615120593
 4,6159 4,6151409 4,615120794

4,63 4,614 4,61514
 4,608 4,61517
 4,615155 4,6151206
 4,615155 4,6151208

4,62 4,61508
 4,613
 4,6151212
 4,6151214

4,616

Les résultats obtenus à l'aide de (VI-7) sont également meilleurs que ceux fournis par diverses améliorations de la méthode de Romberg. Reprenons les exemples donnés par Wallick (54). Les colonnes successives concernent le nombre de chiffres significatifs exacts obtenus par la méthode de Romberg habituelle, par la modification de Krasun-Prager et par celle de Rutishauser sur T_n^0 puis par ces deux modifications à la fois sur T_n^1 , par la modification de Krasun-Prager sur T_n^1 et enfin par (VI-7) :

$$\text{Calcul de } I(a) = \int_0^1 \frac{dx}{1+x^a}$$

n												
4	5	5	5	5	5	5					10	
5	6	6	6	6	6	6	5	5	5	5	5	12
6	8	8	8	8	8	8	7	7	7	7	7	14
7	11	11	11	11	11	11	10	10	10	10	10	14
8	13	13	13	13	13	14	12	12	12	12	12	
9	13	13	14	14	13	15	12	12	12	12	12	
10	12	12	13	14	12	15	11	11	13	12	11	
11	12	12	13	14	12		11	11	12	12	11	
12	12	12	14	13	12		11	11	12	12	11	
13							10	10	12	12	10	
14							10	10	12	12	10	

On voit que pratiquement les résultats donnés par VI-7 sont toujours meilleurs que ceux des autres méthodes et que le nombre de chiffres significatifs exacts est une fonction monotone non décroissante de n ce qui n'est pas toujours le cas des méthodes de Romberg.

6.1.5.2. Application à la dérivation numérique

Les quatre algorithmes utilisés au paragraphe 6.1.5.1. s'appliquent à la dérivation numérique. Soit à approcher la dérivée en $x = 0$ de $f(x) = 1/(x-1)$

par $v(h) = \frac{f(h/2) - f(-h/2)}{h}$ pour des pas h en progression géométrique de raison 1/2. On obtient (le résultat vaut -1) :

-1,333

-1,00667

-1,0039

-0,9824

-1,0000000000

-1,00035

-1,01587

-1,00024

-1,0000038

-0,9955

-1,0000000000

-1,0000000000

-1,0000000000

$$H_{k+2}^{(n-1)} \cdot H_k^{(n+1)} + \left[H_{k+1}^{(n)} \right]^2 = H_{k+1}^{(n-1)} \cdot H_{k+1}^{(n+1)}$$

avec $H_0^{(n)} = 1$ et $H_1^{(n)} = f^{(n)}(t)$ pour $n=0,1,\dots$

2°) Forme accélérée

- paramètre constant d'accélération

$$\bar{E}_{2k}(\omega_k, t) = \bar{E}_{2k-2}(\omega_{k-1}, t) - \omega_k \cdot R_k(t)$$

$$\text{avec } \omega_k = \lim_{t \rightarrow \infty} \frac{1}{1 - \frac{\bar{E}'_{2k}(1, t)}{\bar{E}'_{2k-2}(\omega_{k-1}, t)}}$$

$$\bar{E}_0(\omega_0, t) = f(t)$$

- fonction d'accélération

$$\bar{E}_{2k}(\omega_k(t), t) = \bar{E}_{2k-2}(\omega_{k-1}(t), t) - \omega_k(t) \cdot \bar{R}_k(t)$$

$$\bar{E}_0(\omega_0, t) = f(t)$$

Avant de donner quelques applications numériques de ces formes confluentes établissons quelques formules :

$$R_1 = \frac{f'^2}{f''}$$

$$R_2 = \frac{[f' f''' - f''^2]^2}{f'' [f'' f^{(4)} - f'''^2]^2}$$

$$\omega_1 = \lim_{t \rightarrow \infty} \frac{1}{2 - \frac{f' f'''}{f''^2}}$$

$$\bar{E}'_2(\omega_1, t) = \frac{f'}{f''^2} \left[(1 - 2\omega_1) f''^2 + \omega_1 f' f''' \right]$$

$$R'_2 = \frac{[f'f''' - f'^2] \cdot 2f'' [f'f^{(4)} - f'f'''] [f''f^{(4)} - f''^2] - [f'f''' - f'^2] [f''^2 + f''^2 f^{(5)}]}{f''^2 [f''f^{(4)} - f''^2]^2}$$

$$\omega_2 = \lim_{t \rightarrow \infty} \frac{\bar{E}'_2(\omega_1, t)}{R'_2}$$

On voit, qu'en pratique, il est difficile de continuer. Il est cependant possible d'automatiser le calcul des $R_k(t)$ pour une valeur donnée de t en utilisant la relation de récurrence des déterminants de Hankel. Il est malheureusement impossible d'employer une méthode analogue pour le calcul des ω_k . On est forcé de prendre toujours $\omega_k=1$ ou alors d'utiliser une solution mixte qui consiste à avoir :

$$\omega_k(t) = \frac{1}{1 - \frac{\bar{E}'_{2k}(1, t)}{\bar{E}'_{2k-2}(\omega_{k-1}(t), t)}} = \frac{\bar{E}'_{2k-2}(\omega_{k-1}(t), t)}{R'_k}$$

ce qui donne

$$\omega_1(t) R_1 = \frac{f'^2 f''}{2f''^2 - f' f'''} = S$$

On obtient ainsi pour $\bar{E}'_2(\omega_1(t), t)$ un résultat identique à celui du théorème IV-6. On doit en effet avoir :

$$S = f - \frac{f'^2 f''}{2f''^2 - f' f'''} = 0$$

d'où avec les notations déjà employées :

$$-2g g''^2 + g g' g''' + g'' g'^2 = 0$$

divisons par $g g' g''$ et intégrons, il vient :

$$g g'' = C_1 g'^2$$

dont la solution est $g(t) = C_1 e^{c_2 t}$ ou $\left[c_1 t + c_2 \right]^{c_3}$

La encore, en pratique, il est difficile d'aller plus loin que le premier terme. C'est pour cette raison que nous nous contenterons, presque toujours, de prendre $\omega_k = 1$.

6.2.1. Limite d'une fonction

La principale des applications de la première forme confluyente de l' ϵ -algorithme est le calcul de $\lim_{t \rightarrow \infty} f(t)$. Comme nous l'avons vu, il est toujours possible, par un changement de variable, de se ramener à la limite en un point à droite ou à gauche.

Soit par exemple à calculer $\lim_{x \rightarrow 0} \frac{\sin x}{x}$. Effectuons le changement de variable $t = 1/x$. Nous obtenons $\lim_{t \rightarrow \infty} t \cdot \sin \frac{1}{t}$, d'où

$$f' = \sin \frac{1}{t} - \frac{1}{t} \cos \frac{1}{t}$$

$$f'' = -\frac{1}{t^3} \sin \frac{1}{t}$$

$$f''' = \frac{3}{t^4} \sin \frac{1}{t} + \frac{1}{t^5} \cos \frac{1}{t}$$

$$f^{(4)} = -\frac{12}{t^5} \sin \frac{1}{t} - \frac{8}{t^6} \cos \frac{1}{t} + \frac{1}{t^7} \sin \frac{1}{t}$$

Nous avons obtenu les résultats suivants :

t	R_1	R_2	f	ϵ_2	ϵ_4
1	$-1,078 \cdot 10^{-1}$	$-2,699 \cdot 10^{-2}$	0,84147098	0,94926147	0,97625613
2	$-2,755 \cdot 10^{-2}$	$-6,863 \cdot 10^{-3}$	0,95885107	0,98640309	0,99326671
5	$-4,439 \cdot 10^{-3}$	$-1,109 \cdot 10^{-3}$	0,99334663	0,99778517	0,99889407
10	$-1,111 \cdot 10^{-3}$	$-2,776 \cdot 10^{-4}$	0,99833415	0,99944488	0,99972251
20	$-2,777 \cdot 10^{-4}$	$-6,943 \cdot 10^{-5}$	0,99958336	0,99986110	0,99993052
100	$-1,111 \cdot 10^{-5}$	$-2,776 \cdot 10^{-6}$	0,99998330	0,99999441	0,99999718

On voit, sur ce tableau, d'une part l'amélioration apportée par $\varepsilon_2(t)$ et $\varepsilon_4(t)$ par rapport à $f(t) = t \cdot \sin \frac{1}{t}$ et, d'autre part, que l'erreur sur $\varepsilon_2(t)$ est de l'ordre de grandeur de R_1 et que celle sur $\varepsilon_4(t)$ est de l'ordre de grandeur de R_2 .

Calculons maintenant $\lim_{t \rightarrow \infty} \text{Arctg } t$. Nous trouvons :

$$f' = \frac{1}{1+t^2} \quad f'' = -\frac{2t}{(1+t^2)^2} \quad f''' = -\frac{1-3t^2}{(1+t^2)^3}$$

d'où $\lim_{t \rightarrow \infty} \frac{f' f'''}{f''^2} = \frac{3}{2}$ et $\omega_1 = 2$, ce qui donne

$$\frac{\pi}{2} \sim \text{Arctg } t + \frac{1}{t}$$

qui n'est autre que le début du développement en série de Arctg . On peut donc également utiliser les formes confluentes pour l'estimation de fonctions pour des valeurs élevées des paramètres ou des arguments. On pourra comparer les résultats avec une autre méthode proposée par Wynn (66).

6.2.2. Calcul des intégrales impropres

Prenons $f(t) = \int_a^t g(x) dx$. Nous avons $f^{(n)}(t) = g^{(n-1)}(t)$ pour $n = 1, 2, \dots$. Les $\varepsilon_{2k}(t)$ seront donc des approximations de $\int_a^\infty g(x) dx$ si l'on

suppose cette intégrale convergente. Le second aspect de la question est le suivant : si l'on connaît $\int_a^t g(x) dx$ on peut déduire des $\varepsilon_{2k}(t)$ des approximations de $\int_a^\infty g(x) dx$.

Nous avons ainsi un moyen pour passer de l'intervalle d'intégration $[a, t]$ à l'intervalle $[a, \infty[$ et inversement. C'est une propriété que ne possèdent pas les transformations G de Gray et Atchison (26, 27, 28, 29, 30, 31) qui donnent des approximations de $f(\infty)$ connaissant $f(t)$ et $f(t+k)$ ou $f(t)$ et $f(kt)$.

Donnons un exemple (12); nous avons :

$$\Gamma\left(\frac{1}{2}, x^2\right) = \sqrt{\pi} \cdot \text{erfc } x = \sqrt{\pi} \cdot (1 - \text{erf } x) = \int_{x^2}^{\infty} \frac{e^{-t}}{\sqrt{t}} dt$$

prenons $f(t) = \int_0^t \frac{e^{-x}}{\sqrt{x}} dx$. Nous aurons :

$$\int_0^{\infty} \frac{e^{-t}}{\sqrt{t}} dt = \Gamma\left(\frac{1}{2}, 0\right) = \sqrt{\pi} \neq \varepsilon_{2k}(t)$$

d'où finalement les approximations successives :

$$\operatorname{erf} x \neq 1 + \frac{1}{\sqrt{\pi}} \sum_{i=1}^k R_k = E_k$$

avec :

$$f' = \frac{e^{-t^2}}{t}$$

$$f'' = -\left(\frac{1}{t} + \frac{1}{2t^3}\right) e^{-t^2}$$

$$f''' = \left(\frac{1}{t} + \frac{1}{t^3} + \frac{3}{4t^5}\right) e^{-t^2}$$

les dérivées suivantes étant obtenues grâce à un programme de dérivation formelle. Nous avons obtenu les résultats suivants :

$t = 1$	$R_1 = -0,13836917$	$R_2 = -0,145651176 \cdot 10^{-1}$	$R_3 = -0,30309000 \cdot 10^{-2}$
	$E_1 = 0,86163084$	$E_2 = 0,84706566$	$E_3 = 0,84403476$

$$\operatorname{erf}(1) = 0,84270079$$

$t = 1,5$	$R_1 = -0,32435536 \cdot 10^{-1}$	$R_2 = -0,12909670 \cdot 10^{-2}$	$R_3 = -0,13913150 \cdot 10^{-3}$
	$E_1 = 0,96756448$	$E_2 = 0,96627350$	$E_3 = 0,96613437$

$$\operatorname{erf}(1,5) = 0,96610515$$

$t = 2$	$R_1 = -0,45926636 \cdot 10^{-2}$	$R_2 = -0,79872380 \cdot 10^{-4}$	$R_3 = -0,46709767 \cdot 10^{-5}$
	$E_1 = 0,99540734$	$E_2 = 0,99532747$	$E_3 = 0,99532279$

$$\operatorname{erf}(2) = 0,99532227$$

On observe une très nette amélioration de la précision quand t augmente. On remarque également que R_2 et R_3 sont respectivement du même ordre de grandeur que les erreurs sur E_1 et E_2 et que l'erreur sur E_3 est petite devant R_3 . Si on effectue les calculs analytiques donnant les E_k on s'aperçoit que l'on retrouve exactement les convergents de la fraction continue obtenue par Levy-Soussan (37). Il aurait été également possible de calculer erf t en prenant $f(t) = \int_0^t e^{-x^2} dx$. Cette méthode donne de moins bons résultats. Par exemple pour $t = 2$ on obtient :

$$R_1 = -0,51667466 \cdot 10^{-2} \quad R_2 = 0,57408299 \cdot 10^{-3} \quad R_3 = 0,10437886 \cdot 10^{-3}$$

$$E_1 = 0,99483326 \quad E_2 = 0,99540734 \quad E_3 = 0,99530296$$

La première forme confluyente de l' ϵ -algorithme peut donc être appliquée au calcul de fonctions spéciales. Elle génère, en effet, pour celles-ci des approximations par fractions rationnelles. On peut ensuite, par programme, chercher les coefficients de ces fractions rationnelles.

6.2.3. Résolution des équations

Soit à résoudre $f(x) = 0$. On pose $y = f(x)$ d'où $x = f^{-1}(y)$. Résoudre $f(x) = 0$ revient donc à chercher $\lim_{y \rightarrow 0} f^{-1}(y)$ ou encore, en posant $y = 1/t$, $\lim_{t \rightarrow \infty} f^{-1}(1/t)$. Posons $g(t) = f^{-1}(1/t)$ et appliquons à $g(t)$ la première forme confluyente de l' ϵ -algorithme, on a :

$$\frac{dg}{dt} = \frac{dx}{dt} = \frac{dx}{dy} \cdot \frac{dy}{dt} = - \frac{1}{t^2 f'} \quad \left(f^{(n)} = \frac{d^n f}{dx^n} \right)$$

$$\frac{d^2 g}{dt^2} = \frac{2tf' + t^2 \cdot df'/dt}{t^4 f'^2}$$

avec $\frac{df'}{dt} = \frac{df'}{dx} \cdot \frac{dx}{dt} = - \frac{f''}{t^2 f'}$

$$\text{d'où } \frac{d^2q}{dt^2} = \frac{2tf',^2 - f''}{t^4 f',^3}$$

en continuant on obtient :

$$\omega_1 = \lim_{t \rightarrow \infty} \frac{(2tf',^2 - f'')^2}{2tf',^4 - 2tf',^2 f'' + f' f''' - f''^2} \quad (\text{VI-8})$$

supposons que la racine de $f(x)=0$ soit une racine simple. On a donc $f'(x) \neq 0$ et par conséquent $\omega_1=2$. D'où :

$$\varepsilon_2 = x - 2 \frac{f f'}{2f',^2 - f f''}$$

ε_2 sera donc une meilleure approximation de la racine de $f(x)=0$ que x .

D'où en itérant sur ε_2 la méthode suivante :

$$x_{n+1} = x_n - 2 \frac{f(x_n) \cdot f'(x_n)}{2f',^2(x_n) - f(x_n) \cdot f''(x_n)} \quad (\text{VI-9})$$

On voit que l'on retrouve ainsi une méthode itérative connue: la méthode de Schröder. On sait que cette méthode est d'ordre 3 pour les racines simples et d'ordre 1 pour les racines multiples. En fait pour les racines multiples il ne faut pas prendre $\omega_1 = 2$ mais la valeur donnée par (VI-8). Malheureusement on ne sait pas calculer cette valeur. On peut remédier à cet inconvénient en prenant un ω_1 variable et égal à :

$$\begin{aligned} \omega_1 &= \frac{(2tf',^2 - f'')^2}{2tf',^4 - 2tf',^2 f'' + f' f''' - f''^2} \\ &= \frac{(2 \cdot f',^2 - f f'')^2}{2f',^4 - 2f f',^2 f'' + f^2 f' f''' - f^2 f''^2} \end{aligned}$$

d'où l'itération :

$$x_{n+1} = x_n - \frac{f f' (2f'^2 - f f'')}{2f'^4 - 2f f'^2 f'' + f^2 f' f''' - f^2 f''^2} \quad (\text{VI-10})$$

f et ses dérivées étant calculées au point x_n . Cette méthode présente cependant l'inconvénient de nécessiter le calcul de la dérivée troisième de f .

L' ϵ -algorithme nous permet de plus d'interpréter (VI-9) comme le premier terme du développement asymptotique de $x = f^{-1}(y)$ au voisinage de $y=0$. Cette formule (VI-9) étant d'ordre 3, conduit à des résultats numériques intéressants :

Soit à résoudre $x - e^{-x} = 0$ dont la solution est $x=56714329$. Avec $x_0 = 0$ on obtient :

	méthode de Newton	(VI-9)
x_1	0,506481993	0,57142857
x_2	0,560348181	0,56714329
x_3	0,567067938	"
x_4	0,567143200	"
x_5	0,567143290	"

Soit maintenant à résoudre $(x-1)^6 = 0$. Alors (VI-9) n'est plus que d'ordre 1 et on obtient :

	(VI-9)	(VI-10)
x_0	-2,0000000	-2,0000000
x_1	-1,1429572	-1,0000001
x_2	-0,53061226	"
x_{20}	0,99641440	"

Soit à résoudre $x^6 + 6x^5 + 6x^4 - 18x^3 - 31x^2 - 24x - 36 = 0$ qui a $x = -3$ comme racine double. On obtient $x_0 = -3,5$

	Newton :	(VI-9)	(VI-10)
x_1	-3,28015740	-3,2133565	-2,9242341
x_2	-3,21512193	-3,0804280	-2,9986283
x_3	-3,14092307	-3,0281398	-3,0001018
x_4	-3,09531585	-3,00995401	-3,0000266
x_5	-3,06175829	-3,0031880	-2,9999969
x_{10}	-3,00614008	-2,9997533	-3,0000003
x_{15}	-3,00053279	-3,0004272	-3,0000003

On voit donc que les méthodes (VI-9) et (VI-10) sont toujours au moins aussi bonnes que la méthode de Newton dans les cas les plus défavorables.

6.2.4. Seconde forme discrète de l' \mathcal{E} -algorithme

On peut, à partir de la seconde forme confluente de l' \mathcal{E} -algorithme en obtenir une seconde forme discrète :

$$\mathcal{E}_{-1}^{(n)*} = 0 \quad \text{et} \quad \mathcal{E}_0^{(n)*} = 0 \quad \text{pour } n=0,1,\dots$$

$$\mathcal{E}_{2k+2}^{(n)*} = \mathcal{E}_{2k}^{(n+1)*} + \frac{1}{\mathcal{E}_{2k+1}^{(n+1)*} - \mathcal{E}_{2k+1}^{(n)*}}$$

$$\mathcal{E}_{2k+1}^{(n)*} = \mathcal{E}_{2k-1}^{(n+1)*} + \frac{1}{\mathcal{E}_{2k}^{(n+1)*} - \mathcal{E}_{2k}^{(n)*} + \Delta S_{n+k}}$$

On a les relations suivantes avec la première forme discrète de l' \mathcal{E} -algorithme

$$\mathcal{E}_{2k}^{(n)*} = \mathcal{E}_{2k}^{(n)} - S_{n+k} \quad \text{et} \quad \mathcal{E}_{2k+1}^{(n)*} = \mathcal{E}_{2k+1}^{(n)}$$

```
NPI=0
ICC=0
IAC=0
IBC=0
LB=1
LE=0
M=0
DO 27 I=1,N2
TERM1=0.00
27 E(I)=0.00
1 TERM=TERM/PT
A0=TERM-TERM1
IF(DABS(A0).LT.1.D-300)A0=1.D-300
A1=A0+E(1)
IS=0
TERM1=TERM
2 IF(IS.NE.0)A0=A1-E(IS+1)
IF(DABS(E(IS+1)).LT.1.D-300)GO TO 4
IF(IS.GE.M.OR.ICC.NE.0.OR.DABS(A0/E(IS+1)).GT.ANC)GO TO 4
NPI=1
LE=LE+1
4 IF(LIP.NE.1.AND.NPI.NE.1)GO TO 6
IF(NPI.NE.1)GO TO 8
IF(LIP.NE.1)GO TO 10
IF(IS.NE.IS1(LB))GO TO 10
TERM=TERM*PT
RETURN 6
10 LIP=1
IAC=1
NPI=0
IS1(LE)=IS
8 IF(ICC.NE.1)GO TO 14
PA=D+B(LB)-A(LB)
A0=PA/(1.00+PA/E(IS))
ICC=0
LB=LB+1
IF(LB.GT.LE)LIP=0
GO TO 15
14 BB=0.00
IF(IS.NE.0)BB=E(IS)
IF(DABS(A0).LT.1.D-300)A0=1.D-300
A0=1.00/A0+BB
15 IF(LIP.NE.1)GO TO 21
IF(IAC.NE.1)GO TO 18
BB=0.00
IF(IS.NE.0)BB=E(IS)/(1.00-E(IS)/A0)
A(LE)=BB
IAC=0
IBC=1
GO TO 21
18 IF(IBC.NE.1)GO TO 20
IBC=0
IF(DABS(A1).LT.1.D-300) A1=1.D-300
B(LE)=E(IS+1)/(1.00-E(IS+1)/A1)
GO TO 21
20 IF(IS.EQ.0.OR.IS.NE.(IS1(LB)+1))GO TO 21
D=A1/(1.00-A1/E(IS+1))
```

```
ICC=1
GO TO 21
6 BB=0.D0
  IF (IS.NE.0) BB=E (IS)
  IF (DABS(A0).LT.1.D-300) A0=1.D-300
  A0=1.D0/A0+BB
21 IF (IS.NE.0) E (IS)=A2
  A2=A1
  A1=A0
  IS=IS+1
  IF (M.GE.IS) GO TO 2
  E (M+1)=A2
  E (M+2)=A1
  IF (M.LE.0) GO TO 24
  IF (2*(M/2).EQ.M) GO TO 25
  A2=E (M)
  RES=A2*PT
  IF (M.EQ.1) GO TO 24
  RA1=E (M-2)
  A1=RA2
  GO TO 26
25 A1=A2
  RA2=E (M-1)
  A2=RA1
  RES=A1*PT
26 IF (DABS(A1).LT.1.D-300) GO TO 29
  A2=A2/A1
  PREC=DABS(1.D0=A2)
  GO TO 24
29 PREC=DABS(A2)
24 M=M+1
  TERM=TERM*PT
  IF (M.LT.NMM) RETURN
  RETURN 7
END
```

C
C*****

C DENOMINATION
C SOUS-PROGRAMME RHOAL2

C OBJET
C CALCUL DE LA LIMITE D'UNE SUITE A L'AIDE DU RHO-ALGORITHME

C APPEL
C CALL RHOAL2(TERM,PAR,NC,IKK,RES,PREC,\$,\$)

C ARGUMENTS
C TERM VALEUR D'UN TERME DE LA SUITE ENTREE REEL DOUBLE PREC.
C PAR VALEUR DU PARAMETRE CORRESPONDANT A TERM ENTREE DOUBLE PREC.
C NC NOMBRE DE CHIFFRES SIGNIFICATIFS POUVANT ETRE PERDUS LORS DE LA
C SOUSTRACION DE 2 REELS DOUBLE PRECISION VOISINS#NOMBRE DE
C CHIFFRES EXACTS DESIRES. ENTREE ENTIER
C IKK ARGUMENT DEVANT AVOIR LA VALEUR 0 A CHAQUE PREMIER APPEL DU
C SOUS-PROGRAMME POUR UNE NOUVELLE SUITE. IKK EST MODIFIE PAR
C LE SOUS-PROGRAMME ENTREE ENTIER
C RES VALEUR APPROCHEE DE LA LIMITE SORTIE REEL DOUBLE PREC.
C PREC PRECISION RELATIVE OBTENUE SORTIE REEL DOUBLE PREC.
C \$ ETIQUETTE DE SORTIE EN CAS DE SINGULARITE NON ISOLEE. LA METHODE
C N'EST ALORS PLUS APPLICABLE MAIS ELLE A CONVERGE ET LE NOMBRE DE
C CHIFFRES EXACTS OBTENU EST EGAL A NC
C \$ ETIQUETTE DE SORTIE SI LA SUITE INITIALE A PLUS DE 200 TERMES

C SOUS-PROGRAMMES REQUIS
C NEANT

C CARACTERISTIQUES
C ENCOMBREMENT MEMOIRE= 2396 MOTS
C L'UTILISATION DE CE SOUS-PROGRAMME SE FAIT EN PARALLELE AVEC LE CALCUL
C DES TERMES DE LA SUITE. APRES LE CALCUL DE CHAQUE TERME DE LA SUITE ON
C APPELLE RHOAL2 QUI FOURNIT UNE NOUVELLE VALEUR DE RES ET DE PREC, PUIS
C ON SORT DU SOUS-PROGRAMME POUR CALCULER LE TERME SUIVANT DE LA SUITE.
C A CAUSE DES INITIALISATIONS ET DE L'UTILISATION EN PARALLELE DU
C SOUS- PROGRAMME, IL EST IMPOSSIBLE DE TRAITER SIMULTANEMENT PLUSIEURS
C SUITES. ON NE PEUT LES TRAITER QUE SUCCESSIVEMENT.
C LE NOMBRE MAXIMUM DE TERMES DE LA SUITE EST DE 200

C*****

C SUBROUTINE RHOAL2(TERM,PAR,NC,IKK,RES,PREC,\$,\$)
C DOUBLE PRECISION RES,TERM,BB,PREC,A0,A1,A2,E(202),ANC,A(202),B(202
C 1),D,PA,TERM1,PAR,X(202),RA1,RA2,PT
C DIMENSION IS1(202)
C NMM=200
C N2=NMM+2
C IF(IKK.EQ.1)GO TO 1
C PT=TERM
C IF(DABS(PT).LT.1.D-300)PT=1.D0
C RES=TERM
C PREC=1.D100
C IKK=1

```
ANC=10.00**(-NC)
LIP=0
NPI=0
ICC=0
IAC=0
IBC=0
LB=1
LE=0
M=0
DO 27 I=1,N2
27 E(I)=0.00
TERM1=0.00
1 TERM=TERM/PT
A0=TERM-TERM1
X(M+1)=PAR
A1=A0+E(1)
IS=0
TERM1=TERM
2 IF(IS.NE.0)A0=A1-E(IS+1)
IF(DABS(A0).LT.1.0-300)A0=1.0-300
IF(DABS(E(IS+1)).LT.1.0-300)GO TO 4
IF(IS.GE.M.OR.ICC.NE.0.OR.DABS(A0/E(IS+1)).GT.ANC)GO TO 4
NPI=1
LE=LE+1
4 IF(LIP.NE.1.AND.NPI.NE.1)GO TO 6
IF(NPI.NE.1)GO TO 8
IF(LIP.NE.1)GO TO 10
IF(IS.NE.IS1(LB))GO TO 10
TERM=TERM*PT
RETURN 7
10 LIP=1
IAC=1
NPI=0
IS1(LE)=IS
8 IF(ICC.NE.1)GO TO 14
PA=D+B(LB)-A(LB)
IF(IS.EQ.M)GO TO 28
A0=PA/(X(M+1)-X(M-IS ))+PA/E( IS )
28 ICC=0
LB=LB+1
IF(LB.GT.LE)LIP=0
GO TO 15
14 BB=0.00
IF(IS.NE.0)BB=E( IS )
IF(IS.EQ.M)GO TO 15
IF(DABS(A0).LT.1.0-300)A0=1.0-300
A0=(X(M+1)-X(M-IS ))/A0+BB
15 IF(LIP.NE.1)GO TO 21
IF(IAC.NE.1)GO TO 18
BB=0.00
IF(IS.NE.0)BB=E( IS )/(X(M )-X(M-IS+1))-E( IS )/A0
A(LE)=BB
IAC=0
IBC=1
GO TO 21
18 IF(IBC.NE.1)GO TO 20
IBC=0
```

```
IF(IS.EQ.M)GO TO 21
B(LE)=E(IS+1)/(X(M)-X(M-IS))-E(IS+1)/A1
GO TO 21
20 IF(IS.EQ.0.OR.IS.NE.(IS1(LB)+1))GO TO 21
D=A1/(X(M+1)-X(M-IS+1))-A1/E(IS+1)
ICC=1
GO TO 21
6 BB=0.D0
IF(IS.NE.0)BB=E(IS)
IF(IS.EQ.M)GO TO 21
IF(DABS(A0).LT.1.D-300)A0=1.D-300
A0=(X(M+1)-X(M-IS))/A0+BB
21 IF(IS.NE.0)E(IS)=A2
A2=A1
A1=A0
IS=IS+1
IF(M.GE.IS)GO TO 2
E(M+1)=A2
E(M+2)=A1
IF(M.LE.0)GO TO 24
IF(2*(M/2).EQ.M)GO TO 25
A1=E(M)+(X(M+1)-X(1))/A0
A2=E(M)
RES=A2*PT
IF(M.EQ.1)GO TO 24
RA1=E(M-2)
A1=RA2
GO TO 26
25 A1=A2
RA2=E(M-1)
A2=RA1
RES=A1*PT
26 IF(DABS(A1).LT.1.D-300)GO TO 29
A2=A2/A1
PREC=DABS(1.D0-A2)
GO TO 24
29 PREC=DABS(A2)
24 M=M+1
TERM=TERM*PT
IF(M.LT.NMM)RETURN
RETURN 8
END
```

DENOMINATION

SOUS-PROGRAMME EPC

OBJET

CALCUL DE LA LIMITE D'UNE FONCTION A L'AIDE DE LA PREMIERE FORME
CONFLUENTE DE L'EPSILON-ALGORITHME

APPEL

CALL EPC(KM,F,T,E,R)

ARGUMENTS

KM NOMBRE DE TRANSFORMATIONS A EFFECTUER SUR LA FONCTION INITIALE.
LE SOUS-PROGRAMME APPLIQUE 2*KM FOIS L'ALGORITHME ENTREE ENTIER

F FONCTION FORTRAN DOUBLE PRECISION QUI CALCULE LA FONCTION ET SES
2*KM PREMIERES DERIVEES AU POINT D'ABSCISSE T. LE PREMIER
ARGUMENT DE CETTE FONCTION EST I QUI EST L'ORDRE DE DERIVATION
(I=0 CALCUL DE LA FONCTION ELLE MEME). LE SECOND ARGUMENT EST
T VALEUR DE LA VARIABLE.

T VALEUR DE LA VARIABLE ENTREE D.P.

E VECTEUR DES ESTIMATIONS SUCCESSIVES DE LA LIMITE,C'EST A DIRE
VALEURS DE EPSILON(2*K) POUR K=1,000,KM SORTIE DOUBLE PREC.

R VECTEUR DES QUANTITES R. R(I),I=2,KM EST L'ERREUR ABSOLUE SUR
E(I-1). R(1) EST L'ERREUR SUR F(0,T) EN TEMPS QU'ESTIMATEUR DE
LA LIMITE. SORTIE DOUBLE PREC.

SOUS-PROGRAMMES REQUIS

FONCTION FORTRAN F A ECRIRE PAR L'UTILISATEUR

CARACTERISTIQUES

ENCOMBREMENT MEMOIRE = 339 MOYS
KM DOIT ETRE INFERIEUR OU EGAL A 15.
E ET R DOIVENT ETRE DIMENSIONNES A KM MINIMUM DANS LE PROGRAMME
APPELANT. F DOIT ETRE DECLAREE EN EXTERNAL ET EN DOUBLE
PRECISION DANS LE PROGRAMME APPELANT.

SUBROUTINE EPC(KM,F,T,E,R)

DOUBLE PRECISION E(1),R(1),H(3,31),F,T

MK=KM

IF(KM.LE.15)GO TO 6

MK=15

PRINT 5

5 FORMAT(/' PREMIER ARGUMENT RAMENE A 15'/)

6 M=2*MK+1

DO 2 I=1,M

H(1,I)=1.D0

2 H(2,I)=F(I,T)

R(1)=H(2,2)*H(2,2)/H(2,3)

DO 3 K=1,MK

M=M+2

DO 4 I=1,M

4 H(3,I)=(H(2,I)*H(2,I+2) - H(2,I+1)**2)/H(1,I+2)

```
E(K)=H(3,1)/H(2,3)
IF(K.EQ.MK)RETURN
R(K+1)=H(3,2)*H(3,2)/H(3,3)/H(2,3)
DO 3 I =1, M
H(1, I)=H(2, I)
3 H(2, I)= H(3, I)
END
```

Les programmes précédents ont été testés sur un ordinateur UNIVAC 1108 dont les caractéristiques sont les suivantes :

Cycle mémoire : 0,75 μs

mot en double précision: 72 bits \longrightarrow

mantisse	60
exposant	11
signe	1

durée des opérations arithmétiques élémentaires (en double précision) :

addition - soustraction : entre 2,625 et 3,375 μs

multiplication " 4,25 et 5 μs

division " 17,25 et 18 μs

On a les temps suivants (en secondes et à $\pm 20\%$ près)

Nombre de termes de la suite	EPSAL 2	RHOAL 2
10	0,015	0,015
20	0,040	0,043
30	0,078	0,091
50	0,207	0,236
70	0,433	0,475
100	0,829	0,960
120	1,180	1,371
150	1,807	2,137
170	2,293	2,724
200	3,101	3,755

B I B L I O G R A P H I E

- 1 - AITKEN (A.C.)
determinants and matrices
Oliver and Boyd - 7 ED. 1951 -
- 2 - AITKEN (A.C.)
on bernouilli's numerical solution of algebraic equations
Proc. Roy. Soc. EDINBURGH , vol 46, pp 289-305 -1926-
- 3 - BANACH (S)
théorie des opérations linéaires
Chelsea Publ. Comp. NEW YORK - 1955 -
- 4 - BARANGER (J)
une formule optimale pour le calcul de la somme d'une série
Colloque d'Analyse numérique du C.N.R.S. - SUPER-BESSE - Juin 1970 -
- 5 - BAUER (F.L.)
nonlinear sequence transformations
approximation of functions - 1965 -
- 6 - BAUER (F.L.)
the g-algorithm
J. Soc. Indust. Appl. Math. vol 8 N° 1 - March 1960 -
- 7 - BOURBAKI (N)
fonctions d'une variable réelle (chap. 5)
Hermann n° 1132 . PARIS 1951 -
- 8 - BREZINSKI (C)
application de l' \mathcal{E} -algorithme à la résolution des systèmes non
linéaires - C.R.A.S. Ser. A T.271. pp 1174-1177 -9 Décembre 1970 -
- 9 - BREZINSKI (C)
application du \mathcal{F} -algorithme à la quadrature numérique
C.R.A.S. PARIS. T270 . Ser. A pp 1252-1253 - 11 Mai 1970 -
- 10 - BREZINSKI (C)
études sur les \mathcal{E} et \mathcal{F} -algorithmes
A paraître dans numér. Math.
- 11 - BREZINSKI (C)
les \mathcal{E} et \mathcal{F} -algorithmes
Colloque d'Analyse numérique du C.N.R.S. SUPER-BESSE - Juin 1970 -

- 12 - BREZINSKI (C)
méthodes non linéaires d'accélération de la convergence
Séminaire d'Analyse Numérique - GRENOBLE - 14 Janvier 1970 -
- 13 - BREZINSKI (C)
résultats sur les procédés de sommation et l' ϵ -algorithme
RIRO. 4ième année, R 3, pp 147-153 - 1970 -
- 14 - BREZINSKI (C)
sur un algorithme de résolution des systèmes non linéaires
C.R.A.S. Ser. A T.272 pp145-148 - 11 Janvier 1971 -
- 15 - BREZINSKI (C). CROUZEIX (M)
remarques sur le procédé Δ^2 d'AITKEN
C.R.A.S. PARIS T.270. Ser. A pp 896-898 - 6 Avril 1970 -
- 16 - CLARK (W.D.)
infinite series transformations and their applications
Doctoral Thesis, AUSTIN Texax, -1967 -
- 17 - CLARK (W.D.), GRAY (H.L.), ADAMS (J.E.)
a note on the T-Transformation of LUBKIN
J. research NBS vol 73b n° 1 pp 25-29 - 1969 -
- 18 - COLLATZ (L)
functional analysis and numerical mathematics
Academic Press 1966 -
- 19 - DANIEL (J.W)
summation of series of positive terms by condensation
transformation . Math. Comp. vol 23 n° 105 pp 91-96 - 1969 -
- 20 - DIEUDONNE (J)
calcul infinitesimal
HERMANN - PARIS - 1968 -
- 21 - DIEUDONNE (J)
fondements de l'analyse moderne
GAUTHIER-VILLARS - PARIS - 1967 -
- 22 - DUNFORD (N), SCHWARTZ (J.T.)
linear operators
Interscience Publ. - 1958 -

- 23 - GARSOUX (J)
espaces vectoriels topologiques et distributions
DUNOD - PARIS - 1963 -
- 24 - GASTINEL (N)
analyse numérique linéaire
HERMANN - 1966 -
- 25 - GRAGG (W.B.)
on extrapolation algorithms for ordinary initial value problems
SIAM Numer. Anal. vol 2 n° 3. pp 367-403 - 1965 -
- 26 - GRAY (H.L.), ATCHISON (T.A.)
anote on the G-transformation
Journal of Research NBS vol 72 B n°1 - 1968 -
- 27 - GRAY (H.L.) , ATCHISON (T.A.)
applications of the G and B transforms to the LAPLACE
transform. Proceedings ACM National conference - 1968 -
- 28 - GRAY (H.L.) , ATCHISON (T.A.)
nonlinear transformations related to the evaluation of improper
integrals - SIAM J. Numer. Anal. vol 4 n°3 part. 1 1967 -
vol 5 n°2 part. 2 - 1968 -
- 29 - GRAY (H.L.), ATCHISON (T.A.)
the generalized G-transform.
Math. of Comp. vol 22 n°103 - pp 595-606 - 1968 -
- 30 - GRAY (H.L.), CLARK (W.D.)
on a class of nonlinear transformations and their application
to the evaluation of infinitie series
J. Res. NBS vol 73 B n° 0 pp251-274 - 1969 -
- 31 - GRAY (H.L.), SCHUCANY (W.R.)
some limiting cases of the G-transformation
Math. of Comp. vol 23 n° 108 - October 1969 -
- 32 - GUELFOND (A.O.)
calcul des différences finies
DUNOD - 1963 -
- 33 - HARDY (G.H.)
divergent series
OXFORG Clarendon Press - 1949 -

- 34 - KANTAROVITCH (L.V.), AKILOV (G.P.)
functional analysis in normed spaces
Pergamon Press - 1964 -
- 35 - LAURENT (P.J.)
cours de théorie de l'approximation
Faculté des Sciences de - GRENOBLE -
- 36 - LAURENT (P.J.)
étude des procédés d'extrapolation en analyse numérique
Thèse - GRENOBLE - 1964 -
- 37 - LEVY-SOUSSAN (G)
application des fractions continues à la programmation de
quelques fonctions remarquables
Thèse 3ième Cycle - GRENOBLE - 1962 -
- 38 - LUBKIN (S)
a method of summing infinite series
J. Res. NBS vol 48 n° 3 pp 228-254 - Mars 1952 -
- 39 - MARX (I)
remark concerning a non-linear sequence-to-sequence transform
J. Math. Phys. vol 42 n° 4 pp 334-335 - Décembre 1963 -
- 40 - MAURIN (K)
methods of Hilbert spaces
PWN - VARSOVIE - 1967 -
- 41 - MCLEOD (J.B.)
a fundamental result for the \mathcal{E} -algorithm
MRC - TSR 685 -
- 42 - MIKELADZE (SH.E.)
numerical methods of mathematical analysis
AEC - TR - 4285 -
- 43 - MILNE-THOMSON
the calculus of finite differences
Mac Millan - 1933 -
- 44 - OSTROWSKI (A.M.)
solution of equations and systems of equations
Academic Press - 1966 -

- 45 - OVERHOLT (K.J.)
extended ALTKEN acceleration
Bit. vol 5 pp 122-132 - 1965 -
- 46 - RICHARDSON (L.F.)
the deferred approach to the limit
Phil. Trans. Roy. Soc. vol 226 pp 299-261 - LONDON - 1927 -
- 47 - RIESZ (F), NAGY (B.SZ)
leçons d'analyse fonctionnelle
Académie des Sciences de HONGRIE - 1952 -
- 48 - RITUSHAUSER (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 pp 369-383 - 1951 -
- 50 - SCHWARTZ (L)
cours d'analyse
Ecole Polytechnique - PARIS - 1966 -
- 51 - SHANKS (D)
non linear transformations of divergent and slowly convergent
series - J. Math. Phys. vol. 34 n°1 - pp 1-42 - Avril 1955 -
- 52 - TREVES (F)
éléments de la théorie des espaces topologiques
SEDES - PARIS 1959 -
- 53 - ULM (S.YU)
extension of STEFFENSEN'S method for solving nonlinear operator
equations - U.S.S.R. Comp. Math. Phys. vol. 4 n°6 pp 159-165
- 54 - WALLICK (G.C.)
Remark on algorithm 351. modified ROMBERG quadrature
Comm. ACM vol. 13 n°6 - Juin 1970 -
- 55 - WIJNGAARDEN (A.VAN.)
numerical analysis as an independent science
Bit. vol. 6 pp 66-81 - 1966 -

- 56 - WYNN (P)
a general system of orthogonal polynomials
Quart. J. Math. ser. 2 vol. 18 - pp 81-69 - OXFORD 1967 -
- 57 - WYNN (P)
a note on a confluent form of the ϵ -algorithm
Arch. Math. vol. 11 pp 237-240 - 1960 -
- 58 - WYNN (P)
a note on programming repeated application of the ϵ -algorithm
chiffres vol. 8 n°1 pp 23-62 - 1965 -
- 59 - WYNN (P)
acceleration techniques for iterated vector and matrix
problems - Math. of Comp. vol. 16 n°79 pp 301-322 - Jul. 1962
- 60 - WYNN (P)
acceleration techniques in numerical analysis with particular
reference to problems in one independent variable - Proc.
IFIP . pp 149-156 - North HOLLAND Pub. Co - Congress 1962 -
- 61 - WYNN (P)
an arsenal of algol procedures for the evaluation of continued
fractions and for effecting the epsilon algorithm - RFTI
chiffres vol. 9 n°4 - 1966 -
- 62 - WYNN (P)
confluent forms of certain non linear algorithms
Arch. Math. vol. 11 pp 223-236 - 1960 -
- 63 - WYNN (P)
on a connection beetwen the first and the second confluent
form of the ϵ -algorithm
NIEUW. Arch. Wisk. (3) vol. 11 pp 19-21 - 1963 -
- 64 - WYNN (P)
on a device for computing the $e_{m,n}(S_n)$ transformation - MTAC
vol. 10 n°54 pp 91-96 - Avril 1956 -
- 65 - WYNN (P)
on a procrustean technique for the numerical transformation
of slowly convergent sequences and series - Proc. vol. 52
n° 4 - CANBRIDGE Philosophical Society - October 1956 -
- 66 - WYNN (P)
on the computation of certain functions of large argument and
parameter Bit. vol 6 pp 228-259 - 1966 -

- 67 - WYNN (P)
on the convergence and stability of the epsilon algorithm
J. Siam Numerical Anal. vol. 3 pp 91-122 - 1966 -
- 68 - WYNN (P)
on the propagation of error in certain non linear algorithms
Numerische Mathematik vol. 1 n°3 pp 142-149 - July. 1959 -
- 69 - WYNN (P)
partial differential equations associated with certain non-
linear algorithms - Zamp vol. 13 n°3 - 1964 -
- 70 - WYNN (P)
singular rules for certain non-linear algorithms
Nord. Tid. Inf. Beh vol. 3 - 1963 -
- 71 - WYNN (P)
the rational approximation of functions which are formally
defined by a power series expansion - Maths. of Comp. vol 14
n°70 - April 1960 -
- 72 - WYNN (P)
upon a conjecture concerning a method for solving linear
equations and certain other matters - MRC Technical summary
report n° 626 - UNIV. OF WISCONSIN - 1966 -
- 73 - WYNN (P)
upon a second confluent form of the ϵ -algorithm
Proc. Glasg. Math. Soc. vol 5 pp 160-165 - 1962 -
- 74 - WYNN (P)
upon systems of recursions which obtain among the quotients
of the padé table. numer. Math. vol. 8 n° 3 - 1966 -
- 75 - YOSIDA (K)
functional analysis
SPRINGER VERLAG - 2ED. - 1968 -
- 76 - ZAMANSKY (M)
introduction à l'algèbre et à l'analyse modernes
DUNOD - 1967 -

VU

Grenoble, le

Le Président de la Thèse

VU, et permis d'imprimer

Grenoble, le

Le Président de l'Université Scientifique
et Médicale.

