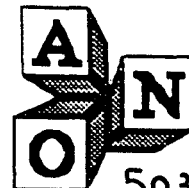


143032

USTL

LABORATOIRE D'ANALYSE NUMERIQUE ET
D'OPTIMISATION



50376
1997
473

n° d'ordre : 2147

THESE

Nouveau régime

présentée à
l'Université des Sciences et Technologies de Lille

pour obtenir le titre de
DOCTEUR en MATHÉMATIQUES
par

Olivier RENAULT



ALGORITHME D'EUCLIDE, POLYNOMES ORTHOGONAUX ET APPROXIMANTS DE PADE

soutenue le 18 décembre 1997 devant la commission d'examen

Membres du jury

Président : C. BREZINSKI, Professeur, Un. Lille 1
Rapporteurs : A. BULTHEEL, Professeur, Leuven, Belgique
J. DELLA DORA, Professeur, IMAG, Grenoble
J. GILEWICZ, Professeur, Un. Toulon
Membres : B. BECKERMANN, Maître de Conférence, Un. Lille
P. WITOMSKI, Professeur, IMAG, Grenoble

DOYENS HONORAIRES DE L'ANCIENNE FACULTE DES SCIENCES

M. H. LEFEBVRE, M. PARREAU

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

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

PROFESSEUR EMERITE

M. A. LEBRUN

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

MM. M. PARREAU, J. LOMBARD, M. MIGEON, J. CORTOIS, A. DUBRULLE

PRESIDENT DE L'UNIVERSITE DES SCIENCES ET TECHNOLOGIES DE LILLE

M. P. LOUIS

PROFESSEURS - CLASSE EXCEPTIONNELLE

M. CHAMLEY Hervé	Géotechnique
M. CONSTANT Eugène	Electronique
M. ESCAIG Bertrand	Physique du solide
M. FOURET René	Physique du solide
M. GABILLARD Robert	Electronique
M. LABLACHE COMBIER Alain	Chimie
M. LOMBARD Jacques	Sociologie
M. MACKE Bruno	Physique moléculaire et rayonnements atmosphériques

M. MIGEON Michel
M. MONTREUIL Jean
M. PARREAU Michel
M. TRIDOT Gabriel

EUDIL
Biochimie
Analyse
Chimie appliquée

PROFESSEURS - 1ère CLASSE

M. BACCHUS Pierre
M. BLAYS Pierre
M. BILLARD Jean
M. BOILLY Bénoni
M. BONNELLE Jean Pierre
M. BOSCOQ Denis
M. BOUGHON Pierre
M. BOURIQUET Robert
M. BRASSELET Jean Paul
M. BREZINSKI Claude
M. BRIDOUX Michel
M. BRUYELLE Pierre
M. CARREZ Christian
M. CELET Paul
M. COEURE Gérard
M. CORDONNIER Vincent
M. CROSNIER Yves
Mme DACHARRY Monique
M. DAUCHET Max
M. DEBOURSE Jean Pierre
M. DEBRABANT Pierre
M. DECLERCQ Roger
M. DEGAUQUE Pierre
M. DESCHEPPER Joseph
Mme DESSAUX Odile
M. DHAINAUT André
Mme DHAINAUT Nicole
M. DJAFARI Rouhani
M. DORMARD Serge
M. DOUKHAN Jean Claude
M. DUBRULLE Alain
M. DUPOUY Jean Paul
M. DYMENT Arthur
M. FOCT Jacques Jacques
M. FOUQUART Yves
M. FOURNET Bernard
M. FRONTIER Serge
M. GLORIEUX Pierre
M. GOSSELIN Gabriel
M. GOUDMAND Pierre
M. GRANELLE Jean Jacques
M. GRUSON Laurent
M. GUILBAULT Pierre
M. GUILLAUME Jean
M. HECTOR Joseph
M. HENRY Jean Pierre
M. HERMAN Maurice
M. LACOSTE Louis
M. LANGRAND Claude

Astronomie
Géographie
Physique du Solide
Biologie
Chimie-Physique
Probabilités
Algèbre
Biologie Végétale
Géométrie et topologie
Analyse numérique
Chimie Physique
Géographie
Informatique
Géologie générale
Analyse
Informatique
Electronique
Géographie
Informatique
Gestion des entreprises
Géologie appliquée
Sciences de gestion
Electronique
Sciences de gestion
Spectroscopie de la réactivité chimique
Biologie animale
Biologie animale
Physique
Sciences Economiques
Physique du solide
Spectroscopie hertzienne
Biologie
Mécanique
Métallurgie
Optique atmosphérique
Biochimie structurale
Ecologie numérique
Physique moléculaire et rayonnements atmosphériques
Sociologie
Chimie-Physique
Sciences Economiques
Algèbre
Physiologie animale
Microbiologie
Géométrie
Génie mécanique
Physique spatiale
Biologie Végétale
Probabilités et statistiques

PROFESSEURS - 2ème CLASSE

M. ABRAHAM Francis	Composants électroniques
M. ALLAMANDO Etienne	Biologie des organismes
M. ANDRIES Jean Claude	Analyse
M. ANTOINE Philippe	Génétique
M. BALL Steven	Biologie animale
M. BART André	Génie des procédés et réactions chimiques
M. BASSERY Louis	Géographie
Mme BATTIAU Yvonne	Systèmes électroniques
M. BAUSIERE Robert	Mécanique
M. BEGUIN Paul	Physique atomique et moléculaire
M. BELLET Jean	Physique atomique, moléculaire et du rayonnement
M. BERNAGE Pascal	Sciences Economiques
M. BERTHOUD Arnaud	Sciences Economiques
M. BERTRAND Hugues	Analyse
M. BERZIN Robert	Physique de l'état condensé et cristallographie
M. BISKUPSKI Gérard	Algèbre
M. BKOUCHE Rudolphe	Biologie végétale
M. BODARD Marcel	Biochimie métabolique et cellulaire
M. BOHIN Jean Pierre	Mécanique
M. BOIS Pierre	Génie civil
M. BOISSIER Daniel	Spectrochimie
M. BOIVIN Jean Claude	Physique
M. BOUCHER Daniel	Biologie appliquée aux enzymes
M. BOUQUELET Stéphane	Gestion
M. BOUQUIN Henri	Chimie
M. BROCARD Jacques	Paléontologie
Mme BROUSMICHE Claudine	Mécanique
M. BUISINE Daniel	Biologie animale
M. CAPURON Alfred	Géographie humaine
M. CARRE François	Chimie organique
M. CATTEAU Jean Pierre	Sciences Economiques
M. CAYATTE Jean Louis	Electronique
M. CHAPOTON Alain	Biochimie structurale
M. CHARET Pierre	Composants électroniques optiques
M. CHIVE Maurice	Informatique théorique
M. COMYN Gérard	Composants électroniques et optiques
Mme CONSTANT Monique	Psychophysiologie
M. COQUERY Jean Marie	Sciences Economiques
M. CORIAT Benjamin	Paléontologie
Mme CORSIN Paule	Physique nucléaire et corpusculaire
M. CORTOIS Jean	Chimie organique
M. COUTURIER Daniel	Tectonique géodynamique
M. CRAMPON Norbert	Biologie
M. CURGY Jean Jacques	Physique théorique
M. DANGOISSE Didier	Analyse
M. DE PARIS Jean Claude	Composants électroniques et optiques
M. DECOSTER Didier	Electrochimie et Cinétique
M. DEJAEGER Roger	Informatique
M. DELAHAYE Jean Paul	Physiologie animale
M. DELORME Pierre	Sciences Economiques
M. DELORME Robert	Sociologie
M. DEMUNTER Paul	Physique atomique, moléculaire et du rayonnement
Mme DEMUYNCK Claire	Informatique
M. DENEL Jacques	Physique du solide - cristallographie
M. DEPREZ Gilbert	

M. LATTEUX Michel
M. LAVEINE Jean Pierre
Mme LECLERCQ Ginette
M. LEHMANN Daniel
Mme LENOBLE Jacqueline
M. LEROY Jean Marie
M. LHENAFF René
M. LHOMME Jean
M. LOUAGE François
M. LOUCHEUX Claude
M. LUCQUIN Michel
M. MAILLET Pierre
M. MAROUF Nadir
M. MICHEAU Pierre
M. PAQUET Jacques
M. PASZKOWSKI Stéfan
M. PETIT Francis
M. PORCHET Maurice
M. POUZET Pierre
M. POVY Lucien
M. PROUVOST Jean
M. RACZY Ladislas
M. RAMAN Jean Pierre
M. SALMER Georges
M. SCHAMPS Joël
Mme SCHWARZBACH Yvette
M. SEGUIER Guy
M. SIMON Michel
M. SLIWA Henri
M. SOMME Jean
Melle SPIK Geneviève
M. STANKIEWICZ François
M. THIEBAULT François
M. THOMAS Jean Claude
M. THUMERELLE Pierre
M. TILLIEU Jacques
M. TOULOTTE Jean Marc
M. TREANTON Jean René
M. TURRELL Georges
M. VANEECLOO Nicolas
M. VAST Pierre
M. VERBERT André
M. VERNET Philippe
M. VIDAL Pierre
M. WALLART François
M. WEINSTEIN Olivier
M. ZEYTOUNIAN Radyadour

Informatique
Paléontologie
Catalyse
Géométrie
Physique atomique et moléculaire
Spectrochimie
Géographie
Chimie organique biologique
Electronique
Chimie-Physique
Chimie physique
Sciences Economiques
Sociologie
Mécanique des fluides
Géologie générale
Mathématiques
Chimie organique
Biologie animale
Modélisation - calcul scientifique
Automatique
Minéralogie
Electronique
Sciences de gestion
Electronique
Spectroscopie moléculaire
Géométrie
Electrotechnique
Sociologie
Chimie organique
Géographie
Biochimie
Sciences Economiques
Sciences de la Terre
Géométrie - Topologie
Démographie - Géographie humaine
Physique théorique
Automatique
Sociologie du travail
Spectrochimie infrarouge et raman
Sciences Economiques
Chimie inorganique
Biochimie
Génétique
Automatique
Spectrochimie infrarouge et raman
Analyse économique de la recherche et développement
Mécanique

M. LE MAROIS Henri	Vie de la firme
M. LEMOINE Yves	Biologie et physiologie végétales
M. LESCURE François	Algèbre
M. LESENNE Jacques	Systèmes électroniques
M. LOCQUENEUX Robert	Physique théorique
Mme LOPES Maria	Mathématiques
M. LOSFELD Joseph	Informatique
M. LOUAGE Francis	Electronique
M. MAHIEU François	Sciences économiques
M. MAHIEU Jean Marie	Optique - Physique atomique
M. MAIZIERES Christian	Automatique
M. MANSY Jean Louis	Géologie
M. MAURISSON Patrick	Sciences Economiques
M. MERIAUX Michel	EUDIL
M. MERLIN Jean Claude	Chimie
M. MESMACQUE Gérard	Génie mécanique
M. MESSELYN Jean	Physique atomique et moléculaire
M. MOCHE Raymond	Modélisation,calcul scientifique,statistiques
M. MONTEL Marc	Physique du solide
M. MORCELLET Michel	Chimie organique
M. MORE Marcel	Physique de l'état condensé et cristallographie
M. MORTREUX André	Chimie organique
Mme MOUNIER Yvonne	Physiologie des structures contractiles
M. NIAY Pierre	Physique atomique,moléculaire et du rayonnement
M. NICOLE Jacques	Spectrochimie
M. NOTELET Francis	Systèmes électroniques
M. PALAVIT Gérard	Génie chimique
M. PARSY Fernand	Mécanique
M. PECQUE Marcel	Chimie organique
M. PERROT Pierre	Chimie appliquée
M. PERTUZON Emile	Physiologie animale
M. PETIT Daniel	Biologie des populations et écosystèmes
M. PLIHON Dominique	Sciences Economiques
M. PONSOLLE Louis	Chimie physique
M. POSTAIRE Jack	Informatique industrielle
M. RAMBOUR Serge	Biologie
M. RENARD Jean Pierre	Géographie humaine
M. RENARD Philippe	Sciences de gestion
M. RICHARD Alain	Biologie animale
M. RIETSCH François	Physique des polymères
M. ROBINET Jean Claude	EUDIL
M. ROGALSKI Marc	Analyse
M. ROLLAND Paul	Composants électroniques et optiques
M. ROLLET Philippe	Sciences Economiques
Mme ROUSSEL Isabelle	Géographie physique
M. ROUSSIGNOL Michel	Modélisation,calcul scientifique,statistiques
M. ROY Jean Claude	Psychophysiologie
M. SALERNO François	Sciences de gestion
M. SANCHOLLE Michel	Biologie et physiologie végétales
Mme SANDIG Anna Margarette	
M. SAWERYSYN Jean Pierre	Chimie physique
M. STAROSWIECKI Marcel	Informatique
M. STEEN Jean Pierre	Informatique
Mme STELLMACHER Irène	Astronomie - Météorologie
M. STERBOUL François	Informatique
M. TAILLIEZ Roger	Génie alimentaire
M. TANRE Daniel	Géométrie - Topologie
M. THERY Pierre	Systèmes électroniques
Mme TJOTTA Jacqueline	Mathématiques
M. TOURSEL Bernard	Informatique
M. TREANTON Jean René	Sociologie du travail

M. DERIEUX Jean Claude	Microbiologie
M. DERYCKE Alain	Informatique
M. DESCAMPS Marc	Physique de l'état condensé et cristallographie
M. DEVRAINNE Pierre	Chimie minérale
M. DEWAILLY Jean Michel	Géographie humaine
M. DHAMELINCOURT Paul	Chimie physique
M. DI PERSIO Jean	Physique de l'état condensé et cristallographie
M. DUBAR Claude	Sociologie démographique
M. DUBOIS Henri	Spectroscopie hertzienne
M. DUBOIS Jean Jacques	Géographie
M. DUBUS Jean Paul	Spectrométrie des solides
M. DUPONT Christophe	Vie de la firme
M. DUTHOIT Bruno	Génie civil
Mme DUVAL Anne	Algèbre
Mme EVRARD Micheline	Génie des procédés et réactions chimiques
M. FAKIR Sabah	Algèbre
M. FARVACQUE Jean Louis	Physique de l'état condensé et cristallographie
M. FAUQUEMBERGUE Renaud	Composants électroniques
M. FELIX Yves	Mathématiques
M. FERRIERE Jacky	Tectonique - Géodynamique
M. FISCHER Jean Claude	Chimie organique, minérale et analytique
M. FONTAINE Hubert	Dynamique des cristaux
M. FORSE Michel	Sociologie
M. GADREY Jean	Sciences économiques
M. GAMBLIN André	Géographie urbaine, industrielle et démographie
M. GOBLOT Rémi	Algèbre
M. GOURIEROUX Christian	Probabilités et statistiques
M. GREGORY Pierre	I. A. E.
M. GREMY Jean Paul	Sociologie
M. GREVET Patrice	Sciences Economiques
M. GRIMBLot Jean	Chimie organique
M. GUELTON Michel	Chimie physique
M. GUICHAOUA André	Sociologie
M. HAIMAN Georges	Modélisation, calcul scientifique, statistiques
M. HOUDART René	Physique atomique
M. HUEBSCHMANN Johannes	Mathématiques
M. HUTTNER Marc	Algèbre
M. ISAERT Noël	Physique de l'état condensé et cristallographie
M. JACOB Gérard	Informatique
M. JACOB Pierre	Probabilités et statistiques
M. JEAN Raymond	Biologie des populations végétales
M. JOFFRE Patrick	Vie de la firme
M. JOURNAL Gérard	Spectroscopie hertzienne
M. KOENIG Gérard	Sciences de gestion
M. KOSTRUBIEC Benjamin	Géographie
M. KREMBEL Jean	Biochimie
Mme KRIFA Hadjila	Sciences Economiques
M. LANGEVIN Michel	Algèbre
M. LASSALLE Bernard	Embryologie et biologie de la différenciation
M. LE MEHAUTE Alain	Modélisation, calcul scientifique, statistiques
M. LEBFEVRE Yannic	Physique atomique, moléculaire et du rayonnement
M. LECLERCQ Lucien	Chimie physique
M. LEFEBVRE Jacques	Physique
M. LEFEBVRE Marc	Composants électroniques et optiques
M. LEFEBVRE Christian	Pétrologie
Melle LEGRAND Denise	Algèbre
M. LEGRAND Michel	Astronomie - Météorologie
M. LEGRAND Pierre	Chimie
Mme LEGRAND Solange	Algèbre
Mme LEHMANN Josiane	Analyse
M. LEMAIRE Jean	Spectroscopie hertzienne

M. TURREL Georges
M. VANDIJK Hendrik
Mme VAN ISEGHEM Jeanine
M. VANDORPE Bernard
M. VASSEUR Christian
M. VASSEUR Jacques
Mme VIANO Marie Claude
M. WACRENIER Jean Marie
M. WARTEL Michel
M. WATERLOT Michel
M. WEICHERT Dieter
M. WERNER Georges
M. WIGNACOURT Jean Pierre
M. WOZNIAK Michel
Mme ZINN JUSTIN Nicole

Spectrochimie infrarouge et raman

Modélisation, calcul scientifique, statistiques

Chimie minérale

Automatique

Biologie

Electronique

Chimie inorganique

géologie générale

Génie mécanique

Informatique théorique

Spectrochimie

Algèbre

Remerciements

Je tiens à témoigner toute ma gratitude à Claude BREZINSKI, Professeur à l'Université des Sciences et Technologies de Lille et Directeur du laboratoire d'Analyse Numérique et d'Optimisation, pour la confiance et le soutien constants qu'il m'a accordés en dirigeant ma thèse, les conseils et les encouragements qu'il m'a prodigués et plus encore pour la motivation qu'il m'a patiemment transmise.

Un remerciement tout particulier à Adhemar BULTHEEL, Professeur à la Katholieke Universiteit Leuven, pour l'accueil sympathique qu'il m'a réservé, le temps qu'il a consacré à répondre à mes questions et d'avoir accepté d'être rapporteur de ce travail.

Mes remerciements vont aussi à Jean DELLA DORA, Professeur à l'Institut National Polytechnique de Grenoble pour l'attention qu'il a portée à mon travail en temps que rapporteur et ses remarques pertinentes lors des entretiens qu'il a bien voulu m'accorder.

Je remercie également Jacek GILLEVICZ, Professeur à l'université de Toulon pour avoir accepté d'être rapporteur de ce travail. Je lui suis reconnaissant d'avoir apporté d'utiles précisions.

Je suis très honoré de la présence au jury de Patrick WITOMSKI, Professeur à l'Institut National Polytechnique de Grenoble ainsi que de Bernard BECKERMAN, Maître de Conférence à l'Université des Sciences et Technologies de Lille et que je remercie pour les orientations bibliographiques complémentaires qu'il m'a recommandées.

Je tiens à remercier les membres du laboratoire d'Analyse Numérique et d'Optimisation de Lille pour l'ambiance agréable dans laquelle ils m'ont accueilli. Encore merci à Madame Françoise TAILLY qui a facilité mon organisation à distance.

Je remercie mon épouse et complice, Patricia, pour ses patients encouragements et ma fille Leticia pour sa joie communicative.

Merci à mes parents, mes beaux-parents et amis qui par leur disponibilité m'ont aidé à concilier ce travail avec ma vie professionnelle et familiale.

Résumé

Cette thèse développe les relations entre les polynômes orthogonaux formels, les approximants de Padé et les systèmes linéaires de Hankel afin d'exploiter une stratégie de calculs régressive. A partir de l'identité de Bezout, le procédé d'interpolation et l'algorithme d'Euclide nous permettent une construction régressive de la récurrence à trois termes.

Du point de vue algébrique, nous interprétons en terme de bordage les déplacements progressifs dans la table des polynômes orthogonaux formels. Inversement, nous étudions les relations de récurrences régressives qui ne font pas intervenir les moments de la fonctionnelle linéaire sous-jacente.

L'algorithme régressif que nous avons généralisé est obtenu soit directement par une modification du prolongement de la fonctionnelle linéaire tronquée soit implicitement par le choix d'un polynôme orthogonal. Ceci nous permet d'établir un lien algorithmique entre les approximants de type Padé et les approximants de Padé.

Du point de vue numérique, nous avons modifié l'algorithme régressif pour éviter les divisions par zéro. Nous proposons d'améliorer les conditionnements des sous-systèmes intermédiaires en exprimant les calculs dans d'autres bases polynômiales comme celles de Newton ou de Tchebychev.

Les méthodes récursives utilisées pour sauter les sous-systèmes mal conditionnés reposent sur la résolution de systèmes linéaires de Sylvester et se ramènent au calcul d'approximants de Padé. Dans ce cas, l'algorithme régressif nous permet d'éviter la résolution des petits sous-systèmes mal conditionnés.

Nous avons exprimé en termes de polynômes orthogonaux formels l'inverse des matrices de Hankel, la formule de Gohberg-Semencul et une identité de type Bezout. Nous construisons enfin des algorithmes régressifs pour des systèmes de Hankel à déplacements puis persymétriques et proposons un algorithme d'Euclide progressif.

Mots clefs

Approximants de Padé, identité de Bezout, interpolation, polynômes orthogonaux formels, algorithme d'Euclide, systèmes de Hankel, récurrences régressives.

Abstract

In this thesis, relations between formal orthogonal polynomials, Padé approximants and Hankel linear systems are given to let us apply a backward computation scheme. From the Bezout's identity, the interpolation process and the Euclidean algorithm are performed to allow us the backward computation of the three-terms recurrence.

From an algebraical point of view, we show that forward moves in the table of formal orthogonal polynomials are coming out from bordering methods. Conversely, we study backward recurrence relations that don't use the moments of the underlying linear functional.

The backward algorithm we have generalized is obtained either directly by a prolongation of the truncated linear functional or implicitly by a choice of an orthogonal polynomial. This let us establish an algorithmic link between Padé approximants and Padé-type approximants.

From a numerical point of view, we have modified the backward algorithm to overcome the divisions by zero. We propose to monitor the condition numbers of the intermediate subsystems in other bases like Newton's or Tchebychev's.

Recursive methods for jumping over badly conditioned subsystems rely on the solution of Sylvester's linear systems and are still Padé approximants computation problems. In this case, the backward algorithm let us overcome the solution of the small badly conditioned subsystems.

we have expressed Inverse Hankel matrices, the Gohberg-Semencul formula and a Bezout's type identity in a formal orthogonal polynomials setting. We have also performed the backward algorithm for Hankel systems with a right hand side displacement structured and for persymmetric Hankel systems and proposed a forward Euclidean algorithm.

Key words

Padé approximants, Bezout's identity, interpolation, formal orthogonal polynomials, Euclidean algorithm, Hankel systems, backward recurrences.

Table des matières

1	Présentation du problème	5
1.1	Introduction	5
1.2	Polynômes orthogonaux formels	6
1.2.1	Introduction aux polynômes orthogonaux formels	6
1.2.2	Définition	6
1.2.3	Polynômes réguliers unitaires	7
1.2.4	Forme bilinéaire associée	7
1.2.5	Récurrance à trois termes	10
1.3	Un problème modèle	11
1.3.1	Les approximants de Padé	12
1.3.2	Méthode s'appuyant sur les polynômes orthogonaux formels	13
1.3.3	Méthode reposant sur les fractions continues	19
1.3.4	Une comparaison numérique	22
1.3.5	Perspectives de recherches	24
1.4	Panorama des applications	25
1.4.1	Codes correcteurs d'erreurs	25
1.4.2	Systèmes de Toeplitz en traitement du signal	27
1.4.3	Schéma synoptique	29
1.5	Problématique	32
2	Diverses présentations algébriques des récurrences	35
2.1	Introduction	35
2.2	La récurrence à trois termes et la méthode de bordage	35
2.3	Les polynômes adjacents et autres relations de récurrence	40
2.4	La table des polynômes orthogonaux	47
2.5	La matrice de Jacobi	51
3	L'algorithme régressif	53
3.1	Introduction	53
3.2	L'algorithme d'Euclide et la relation de récurrence à trois termes	53
3.2.1	Définition	55
3.2.2	Calcul des polynômes associés	56
3.2.3	L'identité de Christoffel-Darboux	56

3.3	Transformée de Fourier discrète (DFT) et convolution cyclique	57
3.3.1	Définition	57
3.3.2	Transformée inverse	58
3.3.3	Intérêt algorithmique de la transformée de Fourier rapide	58
3.3.4	Evaluation et interpolation polynomiale	58
3.3.5	Convolution circulaire	58
3.3.6	Convolution et produit ponctuel	59
3.3.7	Produit de polynômes	59
3.4	Différentes connexions entre les matrices de Toeplitz, les matrices de Hankel et la DFT	60
3.4.1	Produit d'une matrice de Toeplitz sur un vecteur	60
3.4.2	Inversion d'une matrice de Toeplitz triangulaire inférieure	61
3.4.3	Résolution d'un système de Toeplitz Hessenberg inférieur	62
3.4.4	Les matrices de Hankel et la convolution circulaire	63
3.5	Polynômes orthogonaux relativement à une fonctionnelle linéaire cyclique et régulière	64
3.5.1	Le dernier polynôme orthogonal relativement à une fonctionnelle cyclique	64
3.5.2	L'avant-dernier polynôme orthogonal	65
3.6	L'algorithme régressif	68
3.6.1	Stratégie de l'algorithme régressif	68
3.6.2	Exemple dans un corps fini	69
3.6.3	Intérêt de l'algorithme régressif	70
3.6.4	Exemples numériques	70
4	Extensions de l'algorithme régressif	75
4.1	Introduction	75
4.2	Mise en œuvre numérique	75
4.2.1	Stabilité et complexité	76
4.2.2	La DFT et les divisions	77
4.2.3	Comment éviter les divisions par zéro	78
4.3	Autres procédés régressifs	83
4.3.1	Autres prolongements périodiques de la fonctionnelle	83
4.3.2	Prolongements quelconques de la fonctionnelle	84
4.3.3	Choix particuliers du dernier polynôme orthogonal	87
4.3.4	Choix particuliers de l'avant-dernier polynôme orthogonal	89
4.3.5	Autres formes de prolongements de la fonctionnelle	90
4.3.6	Lien entre les approximants de type-Padé et les approximants de Padé	92
4.4	Un tour d'horizon	93
4.4.1	Autour de l'algorithme d'Euclide	93
4.4.2	L'interpolation et l'évaluation polynomiale	95
4.4.3	La division Euclidienne et la représentation polynomiale	95

4.4.4	La base de Newton et l'algorithme d'Euclide	96
4.4.5	Méthode des moments modifiés	99
4.4.6	Conclusions sur l'algorithme régressif	99
5	Algorithme de sauts récursif	101
5.1	Introduction	101
5.2	Définitions et généralités	101
5.2.1	Intérêts algorithmiques de la récurrence	102
5.3	Comparaison numérique d'algorithmes pour le calcul des polynômes orthogonaux	104
5.4	Cas des polynômes orthogonaux non réguliers	106
5.5	Cas des polynômes orthogonaux quasi-irréguliers.	107
5.6	Conditionnement des polynômes orthogonaux	115
5.7	Conclusions	118
6	Application aux systèmes de Hankel et à l'algorithme d'Euclide	119
6.1	Inversion des matrices de Hankel	119
6.2	Méthodes de résolution des systèmes de Hankel	123
6.3	Liaison entre les systèmes de Hankel quelconques et les polynômes orthogonaux.	124
6.4	Systèmes linéaires de type Hankel	125
6.4.1	Systèmes linéaires de Hankel cycliques	125
6.5	De l'algorithme d'Euclide aux systèmes de Hankel	127
6.6	Algorithme régressif pour les systèmes de Hankel	129
6.6.1	Une relation à quatre termes	130
6.7	Application aux systèmes de Hankel en déplacement	130
6.8	Application aux systèmes de Toeplitz symétriques	132
6.9	Application à l'algorithme d'Euclide progressif	135
6.9.1	Introduction	135
6.9.2	Généralités historiques	135
6.9.3	Les évolutions de l'algorithme d'Euclide	136
6.9.4	L'algorithme d'Euclide et les polynômes orthogonaux	138
6.9.5	Algorithme d'Euclide progressif	138
6.9.6	Développements possibles	139

Introduction générale

L'algorithme d'Euclide ([44]) et le calcul des polynômes orthogonaux sont deux facettes d'une même problématique. Pour s'en convaincre, ce travail de thèse présente un algorithme, déjà publié sous la forme d'un article ([91]), qui calcule les polynômes orthogonaux par combinaison de la transformée de Fourier discrète et de l'algorithme d'Euclide. Nous avons porté notre attention sur les liens étroits que cet algorithme développe entre les domaines suivants:

- Polynômes orthogonaux
- Division Euclidienne
- Systèmes linéaires de type Hankel
- Approximants de Padé

Chacune de ces branches a suivi sa propre évolution en apportant son point de vue et son langage tout en rencontrant les mêmes difficultés numériques. Les avancées récentes dans ces domaines sont nombreuses et nécessitent une compréhension transversale. Ce sont très certainement les similitudes entre ces sujets qui ont présidé à ce travail de recherche au travers des publications du laboratoire ANO de Lille. Les principales sources d'informations sur lesquelles se sont ensuite appuyés les développements sont les ouvrages de Draux ([39]), Bini et Pan ([10]) puis Baker et Graves-Morris ([2]). Le langage de base adopté est résolument celui des polynômes orthogonaux tant que le contexte le permet.

Le premier chapitre est consacré aux polynômes orthogonaux formels. Il a pour objectif d'illustrer par divers points de vue la richesse de cette théorie. Nous développons de nombreuses connexions afin de donner une vaste vue d'ensemble et de mieux percevoir les enjeux.

Le second chapitre se focalise sur les aspects algébriques du calcul des polynômes orthogonaux. L'intérêt essentiel de cette partie de l'exposé est de mettre en parallèle les techniques de calculs qui sont intrinsèques aux polynômes orthogonaux et de dégager un premier aspect du calcul régressif.

Les relations qui existent entre la transformée de Fourier Discrète, la convolution et certaines matrices de Toeplitz d'une part, le lien de parenté entre l'algorithme d'Euclide et la récurrence à trois termes d'autre part, forment les bases de l'algorithme régressif que le chapitre trois se propose de reconstituer. Une interprétation nouvelle en termes d'identité de Bezout et d'interpolation permet une compréhension synthétique de l'algorithme régressif.

De nombreuses variantes de l'algorithme régressif sont développées au chapitre quatre et ont permis de cerner les points délicats que soulève la réalisation numérique de l'algorithme ainsi que son intérêt. Nous étudions notamment le comportement de certaines représentations polynômiales face à l'algorithme d'Euclide.

Le chapitre cinq s'est orienté vers les divers aspects numériques du calcul des polynômes orthogonaux. Nous localisons les difficultés dues aux exigences numériques du problème et nous illustrons chaque point par des exemples. De la reformulation de la notion de saut, nous dégageons l'aspect récursif des calculs. Enfin, à travers un certain nombre de résultats numériques, nous évaluons l'ensemble des procédures.

Le dernier chapitre traite des systèmes de Hankel et des applications de l'algorithme régressif obtenues dans ce domaine. Nous rappelons également des résultats connus sous une forme inédite par le biais des polynômes orthogonaux et nous abordons le calcul progressif de l'algorithme d'Euclide.

Les contenus sont illustrés par des exemples et des schémas explicatifs qui complètent l'aspect théorique. Des programmes ont été réalisés en calcul formel avec Mathematica et Maple et en calcul numérique avec Matlab.

Finalement, il en ressort deux principes fondamentaux communs aux domaines abordés:

- La réversibilité des algorithmes
- Une forme de récursivité pour enjamber les étapes instables

NOTATIONS

- $\mathbb{C}[x]$ est l'algèbre des polynômes à coefficients complexes.
- Pour une suite de polynômes $(P_n)_{n \in \mathbb{N}}$ telle que chaque P_n soit de degré n , on note, tant qu'il n'y a pas d'ambiguïté possible,

$$P_n = \sum_{i=0}^n p_i x^i.$$

Il faut cependant comprendre que les coefficients p_i dépendent de n . Lorsque plusieurs polynômes d'une même suite sont utilisés simultanément, des lettres différentes sont utilisées pour leurs coefficients.

- Lorsque l'on calcule des polynômes par des relations de récurrences on peut être amené à modifier leur normalisation, c'est-à-dire qu'ils sont obtenus à un coefficient multiplicateur près. Pour un polynôme P_n de $\mathbb{C}[x]$ la normalisation unitaire correspond à imposer un coefficient dominant égal à 1. La normalisation en zéro correspond au choix d'un coefficient constant égal à 1, ce qui n'est possible que dans le cas où $P_n(0) \neq 0$.
- La matrice carrée identité de dimension n est notée I_n .
- Le vecteur dont toutes les composantes valent 1 est noté $\mathbf{1}$ s'il n'y pas d'ambiguïté sur sa dimension.
- La matrice carrée nulle de dimension n est notée O_n et le Vecteur nul o_k .
- La matrice adjointe de M est notée M^* .
- Les vecteurs de la base canonique de \mathbb{C}^n sont notés $e_k^{(n)}$, le (n) pouvant être omis s'il n'y a pas d'ambiguïté.
- On définit la multiplication par composantes de deux vecteurs comme l'opération induite par le produit de deux matrices diagonales, chacune d'elles ayant pour diagonale les composantes de l'un des vecteurs. A l'inverse par composantes d'un vecteur correspond l'inverse d'une matrice diagonale.
- Une fonction à valeurs réelles ou complexes définie sur un espace de fonctions est appelée fonctionnelle. Nous n'utiliserons que des fonctionnelles définies sur $\mathbb{C}[x]$ qui sont linéaires. Une telle fonctionnelle c est déterminée par la donnée de ses moments, c'est-à-dire des nombres $c(x^i)$ pour $i \in \mathbb{N}$. Néanmoins, dans les applications numériques nous n'employons qu'un nombre fini de moments. Tronquer une fonctionnelle linéaire consiste à ne garder qu'un nombre fini de moments, quitte à la prolonger par la suite si nécessaire.

- Pour n entier i fixé, la fonctionnelle notée $c^{(i)}$ adjacente à la fonctionnelle c est définie par

$$c^{(i)}(x^j) = c(x^{(i+j)}), \text{ pour tout } j \in \mathbb{N}.$$

Autrement dit les moments sont décalés de i rangs.

- La transformée de Fourier discrète introduite au paragraphe (3.3) est notée en abrégé DFT. Cette notation anglo-saxonne est la plus utilisée en traitement du signal. Le célèbre algorithme rapide qui évalue la DFT est noté FFT.
- La stabilité arithmétique d'un algorithme est la sensibilité que l'algorithme présente face aux erreurs provoquées par le calcul en machine de précision finie. Un algorithme pour résoudre un système linéaire $Ax = b$ est dit stable si la solution calculée x_c est la solution exacte d'un système qui est proche de celui que l'on cherche à résoudre. Cela signifie qu'il existe E et η petits par rapport à A et b respectivement tels que $(A + E)x_c = b + \eta$.
- Le conditionnement d'un problème mesure la sensibilité de la solution du problème face aux perturbations sur les données de départ. Cette notion mesure la difficulté intrinsèque du problème indépendamment de l'algorithme. Un algorithme stable appliqué à la résolution d'un système bien conditionné produit une solution proche de la solution exacte.
- Un algorithme pour résoudre un système linéaire est dit faiblement stable pour une classe de matrices si pour toute matrice bien conditionnée A de cette classe et pour chaque b la solution calculée x_c de $Ax = b$ est telle que $\|x - x_c\|/\|x\|$ soit petit. Autrement dit, la stabilité faible traduit la capacité d'un algorithme à fournir une solution avec une faible erreur relative lorsque le problème est bien conditionné.

Chapitre 1

Présentation du problème

1.1 Introduction

Les polynômes orthogonaux formels ne sont pas seulement utilisés en mathématiques, ils sont très largement employés en physique, interviennent implicitement en traitement du signal et intéressent la biologie et la chimie. Les travaux sur le sujet sont donc épars et nombreux. Notre intention n'est pas d'en résumer tous les aspects. Nous avons porté notre attention sur la représentation algébrique et numérique des polynômes orthogonaux formels. Les résultats que nous avons obtenus seront donc également valables pour les polynômes orthogonaux classiques (ie dont les polynômes dérivés sont aussi orthogonaux) et généraux (ie relativement à une mesure positive). Pourtant, une vue panoramique sur les connexions qu'ils impliquent s'impose afin de situer la portée des résultats que nous exposerons. L'objectif de ce premier chapitre est donc d'illustrer différentes formes sous lesquelles on trouve les polynômes orthogonaux formels dans des domaines aussi variés que le calcul formel ou le traitement du signal. La description qui suit ne prétend pas être exhaustive, elle a pour but essentiel de motiver la lecture de ce travail en exposant des points de vues qui concernent aussi bien l'ingénieur que le mathématicien.

Le premier paragraphe propose différentes représentations algébriques des polynômes orthogonaux formels en retraçant quelques liaisons essentielles avec des problèmes d'algèbre linéaire et bilinéaire.

Le second paragraphe décrit sommairement quelques exemples de développements actuels où figurent implicitement les polynômes orthogonaux. Un schéma synoptique permet de s'orienter dans l'ensemble des liens évoqués.

Après cette vue synthétique, le dernier paragraphe se concentre sur les questions actuelles posées au travers de quelques publications et reformulées ici en termes de polynômes orthogonaux formels. Nous en profitons pour donner brièvement un cadre historique au problème du calcul des polynômes orthogonaux.

1.2 Polynômes orthogonaux formels

1.2.1 Introduction aux polynômes orthogonaux formels

Les polynômes orthogonaux formellement par rapport à une fonctionnelle sur $\mathbb{C}[x]$ définis en (1.2.2) généralisent les polynômes orthogonaux classiques au sens où la forme (1.5) bilinéaire symétrique qui décrit l'orthogonalité ne provient plus nécessairement d'une mesure positive. Cependant, la forme bilinéaire utilisée est symétrique et conserve une structure constante sur ses diagonales, ce qui lui confère la propriété remarquable de réduire le procédé d'orthogonalisation de Gram-Schmidt en une récurrence à trois termes (1.7). L'essentiel des propriétés évoquées dans cette section se trouve dans les ouvrages de Draux [39] et de Brezinski [15]. Ces rappels sont accompagnés de brefs aperçus numériques. Ceux-ci illustrent la problématique autour de laquelle s'est agencé notre travail de thèse, à savoir le calcul et la représentation des polynômes orthogonaux formels.

1.2.2 Définition

Soit c une fonctionnelle linéaire sur l'espace $\mathbb{C}[x]$ des polynômes à coefficients complexes. La fonctionnelle est entièrement déterminée par ses moments

$$c(x^i) = c_i, \text{ pour } i = 0, 1, \dots$$

Une famille $(P_n)_{n \in \mathbb{N}}$ de polynômes orthogonaux formels par rapport à la fonctionnelle c vérifie lorsqu'elle existe

$$\begin{cases} c(x^i P_n(x)) = 0, \text{ pour } i = 0, 1, \dots, n-1. \\ P_n \text{ est de degré } n. \end{cases} \quad (1.1)$$

Pour les coefficients du polynôme de degré n ,

$$P_n(x) = \sum_{i=0}^n p_i x^i,$$

cela se traduit par le système linéaire suivant:

$$\begin{pmatrix} c_0 & c_1 & \cdots & c_{n-1} & c_n \\ c_1 & c_2 & \cdots & c_n & c_{n+1} \\ \vdots & \vdots & & \vdots & \vdots \\ c_{n-1} & c_n & \cdots & c_{2n-2} & c_{2n-1} \\ c_n & c_{n+1} & \cdots & c_{2n-1} & c_{2n} \end{pmatrix} \begin{pmatrix} p_0 \\ p_1 \\ \vdots \\ p_{n-1} \\ p_n \end{pmatrix} = - \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ \alpha \end{pmatrix}. \quad (1.2)$$

Les polynômes P_n sont ainsi définis à une constante multiplicative près α ; on peut donc, sans perdre de généralité, supposer que les polynômes P_n sont unitaires.

1.2.3 Polynômes réguliers unitaires

La matrice M_n des moments de la fonctionnelle linéaire c est constituée des coefficients $c_{i,j} = c(x^{i+j})$ pour $i, j = 1, 2, \dots, n-1$

$$M_n = \begin{pmatrix} c_0 & c_1 & \cdots & c_{n-2} & c_{n-1} \\ c_1 & c_2 & \cdots & c_{n-1} & c_n \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ c_{n-2} & c_{n-1} & \cdots & c_{2n-4} & c_{2n-3} \\ c_{n-1} & c_n & \cdots & c_{2n-3} & c_{2n-2} \end{pmatrix}. \quad (1.3)$$

Une matrice de cette forme est appelée matrice de Hankel et fait partie de l'ensemble des matrices symétriques à coefficients complexes. Si on note les polynômes P_n sous la forme

$$P_n(x) = \sum_{i=0}^{n-1} p_i x^i + x^n$$

les conditions d'orthogonalité (1.2.2) s'expriment alors par le système linéaire suivant

$$\begin{pmatrix} c_0 & c_1 & \cdots & c_{n-2} & c_{n-1} \\ c_1 & c_2 & \cdots & c_{n-1} & c_n \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ c_{n-2} & c_{n-1} & \cdots & c_{2n-4} & c_{2n-3} \\ c_{n-1} & c_n & \cdots & c_{2n-3} & c_{2n-2} \end{pmatrix} \begin{pmatrix} p_0 \\ p_1 \\ \vdots \\ \vdots \\ p_{n-1} \end{pmatrix} = - \begin{pmatrix} c_n \\ c_{n+1} \\ \vdots \\ \vdots \\ c_{2n-1} \end{pmatrix}. \quad (1.4)$$

Ainsi, lorsque le déterminant de Hankel H_n vérifie

$$H_n = \begin{vmatrix} c_0 & c_1 & \cdots & c_{n-2} & c_{n-1} \\ c_1 & c_2 & \cdots & c_{n-1} & c_n \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ c_{n-2} & c_{n-1} & \cdots & c_{2n-4} & c_{2n-3} \\ c_{n-1} & c_n & \cdots & c_{2n-3} & c_{2n-2} \end{vmatrix} \neq 0,$$

les coefficients de P_n sont déterminés de manière unique et le polynôme P_n est dit régulier. La matrice des moments M_n est dite fortement régulière si ses mineurs principaux sont tous réguliers, c'est-à-dire $\det(M_i) = H_i \neq 0$ pour $i = 0, \dots, n$. Cette condition assure l'unicité des polynômes unitaires P_i pour $i = 0, \dots, n$. Replaçons clairement la notion d'orthogonalité (1.2.2) dans le contexte des formes bilinéaires.

1.2.4 Forme bilinéaire associée

Dans tout ce qui suit et sauf mention contraire, nous n'utilisons qu'un nombre fini de moments de la fonctionnelle linéaire c . Nous nous plaçons alors dans le cadre des espaces vectoriels de dimension finie et la fonctionnelle c peut être tronquée pour représenter une

forme linéaire sur $\mathbb{C}_n[x]$. Nous maintiendrons cependant l'appellation de fonctionnelle linéaire, le contexte indiquant clairement la dimension utilisée. Lorsque l'on associe des vecteurs à des polynômes et réciproquement, se pose le problème de la signification des indices. Nous avons occasionnellement opté pour une indexation en degré et non en dimension. Nous le mentionnerons lorsque cela sera le cas et la dimension s'obtiendra en ajoutant une unité à l'indice. Hors de ce contexte spécifique, les indices des matrices et des vecteurs indiqueront leurs dimensions.

L'orthogonalité provient usuellement d'une forme bilinéaire. Nous allons étudier à présent cet aspect afin d'expliquer les particularités dues à la symétrie des matrices de Hankel.

Rappelons tout d'abord que si une forme bilinéaire a pour matrice B dans une base, la matrice B' de la forme bilinéaire dans une autre base s'obtient par une relation de congruence $P^T B P = B'$. Tout en conservant les conditions de forte régularité, la fonctionnelle c permet de définir sur $\mathbb{C}_n[x]$ la forme bilinéaire φ symétrique comme suit

$$\varphi(P, Q) = c(PQ),$$

où P et Q sont deux polynômes de $\mathbb{C}_n[x]$. Autrement dit,

$$\varphi(x^i, x^j) = c(x^{i+j}) = c_{i+j}. \quad (1.5)$$

et ceci pour $1 \leq i$ et $1 \leq j$.

La matrice de Gram de la forme bilinéaire exprimée dans la base $\{1, x, x^2, \dots, x^n\}$ n'est autre que la matrice de Hankel M_n . Dès que l'on définit l'orthogonalité à partir d'une fonctionnelle linéaire on se limite implicitement à une forme bilinéaire dont la matrice de Gram, dans une base quelconque de polynômes, est congruente à une matrice de Hankel. Le procédé d'orthogonalisation de Gram-Schmidt appliqué à la famille de polynômes $\{x^i\}_{i \leq n}$ relativement à φ conduit aux relations d'orthogonalité (1.2.2); la famille $(P_k)_{k \leq n}$ est donc φ -orthogonale. Dans le cas fortement régulier, la recherche d'une base de polynômes orthogonaux formels consiste, du point de vue matriciel, à trouver une matrice triangulaire supérieure P qui vérifie

$$P^T M_n P = D_n,$$

où D_n est une matrice diagonale. Les colonnes de P sont alors constituées des coefficients des polynômes orthogonaux formels. Cela revient à factoriser la matrice M_n sous la forme dite de Gauss-Banachiewicz ([39], p. 92). Soit

$$M_n = T^T D T,$$

où T est une matrice triangulaire et D_n une matrice diagonale dont les éléments diagonaux non nuls sont les $\varphi(P_i, P_i)$. Ce n'est rien d'autre que la factorisation LDU appliquée au cas des matrices symétriques fortement régulières ([12]), ce qui traduit précisément qu'il existe une base φ -orthogonale formée de polynômes $(P_i)_{0 \leq i \leq n}$ où chaque P_i est exactement de degré i . D'après le théorème d'existence de bases orthogonales, on sait que pour toute matrice complexe symétrique, il existe une factorisation un peu plus générale avec T seulement carrée inversible.

Cette description est incompatible avec le cas régulier mais non fortement régulier, puisque l'existence d'un élément nul dans la matrice diagonale D_n est contradictoire avec l'inversibilité de M_n et P . Le problème d'irrégularité de certains polynômes orthogonaux formels n'intervient qu'en présence de vecteurs φ -isotropes (c'est-à-dire tels que $\varphi(P_i, P_i) = 0$). En conséquence, si φ est sans vecteurs isotropes (on dit alors que φ est définie) tous les polynômes orthogonaux formels sont réguliers. Cependant, la forte régularité de la fonctionnelle n'impose pas nécessairement que la forme bilinéaire φ soit définie.

Exemple 1

Si nous prenons par exemple $c = \{1, 0, 1, -1, 0, 1, \dots\}$ pour famille de moments, on a alors

$$M_3 = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & -1 \\ 1 & -1 & 0 \end{pmatrix}.$$

Les déterminants des mineurs principaux sont $D_0 = 1, D_1 = 1$ et $D_2 = -2$. La forte régularité est donc assurée jusqu'au degré deux. La résolution des systèmes successifs donne les polynômes orthogonaux formels réguliers,

$$\begin{aligned} P_0(x) &= 1, \\ P_1(x) &= x, \\ P_2(x) &= x^2 + x - 1, \\ P_3(x) &= x^3 + x^2 + x. \end{aligned}$$

Pourtant, $c((x^2 + 2x)^2) = 0$ de sorte que le polynôme $S(x) = x^2 + 2x$ est isotrope. φ n'est donc pas définie.

◇

Pour φ , le fait d'être définie est plus contraignant que d'être fortement régulière. Une première distinction s'effectue naturellement entre les formes qui sont définies et celles qui ne le sont pas. Mais, seul le cas d'un corps de base réel permet d'en tirer parti.

Lorsque les moments appartiennent à \mathbb{R} , la loi d'inertie de Sylvester ([46], p.52) fait apparaître que les formes bilinéaires symétriques définies sont les formes strictement positives ou strictement négatives. C'est la situation des polynômes orthogonaux classiques pour lesquels la fonctionnelle linéaire est donnée par une mesure sur un intervalle réel et dont la fonction poids est positive, non nulle. Cette dernière condition rend la forme bilinéaire associée définie positive et par suite fortement régulière. En dehors de ce cas, on perd la notion de produit scalaire euclidien et du même coup l'assurance de la forte régularité. Remarquons qu'une matrice de Hankel hermitienne est nécessairement réelle. En effet, H est symétrique donc $H = H^T$, et si elle est hermitienne alors $H = H^*$. Ceci

impose que $H^T = H^*$ et donc que H est réelle.

Lorsque les moments appartiennent à \mathbb{C} , la matrice de Gram n'est plus hermitienne. Les matrices symétriques complexes inversibles ne forment plus qu'une seule classe de congruence pour la transposition ([70]:théorème p. 225). La forte régularité ne peut plus s'obtenir par des choix de classes de congruence particulières. Les classifications que l'on peut obtenir pour le rang, l'irrégularité et les autres caractéristiques des matrices de Hankel ([71]) reposent sur la nature de la suite des déterminants des mineurs principaux.

Dans le cas où c n'est pas fortement régulière, on peut néanmoins obtenir une factorisation en regroupant les polynômes par blocs ([39], p. 93). On obtient une matrice D_n diagonale par blocs et les polynômes obtenus dans des blocs différents sont orthogonaux entre eux, mais les polynômes d'un même bloc ne sont pas orthogonaux entre eux. Cela généralise la forme (1.6) dans le sens où les polynômes réguliers donnent des blocs de dimension 1 sur D_n . Les tailles des blocs correspondent aux sauts que l'on effectue dans la récurrence à trois termes (1.2.5).

En généralisant l'approche bilinéaire, on peut observer que le champ des polynômes orthogonaux formels n'est lui-même qu'une ramification de la notion de biorthogonalité développée par exemple dans [18]. Si la matrice des moments de φ n'a plus la structure de Hankel, chaque ligne i de la matrice de Gram organise sa propre forme linéaire L_i de la manière suivante,

$$\forall P_n(x) \in \mathbb{C}[x], L_i(P_n(x)) = \varphi(x^i, P_n(x)).$$

L'orthogonalité devient relative à la famille de fonctionnelles linéaires $\{L_i, i = 1, 2, \dots\}$. La notion de biorthogonalité reste néanmoins en marge de ce travail qui nécessite que la relation à trois termes, la division Euclidienne et la transformée de Fourier soient associées. Pour certaines familles de fonctionnelles, des résultats récents vont dans ce sens. Notamment avec l'orthogonalité vectorielle ([104]) ou encore avec la biorthogonalité définie par une matrice de Toeplitz ([31]).

1.2.5 Récurrence à trois termes

La condition de forte régularité est également remplie, d'après ([39], p.78), tant que

$$c(x^{i-1}P_{i-1}) \neq 0 \text{ pour } i = 1, 2, \dots, n.$$

Dans ces conditions, les polynômes orthogonaux formels P_i , non nécessairement unitaires, suivent alors la relation de récurrence à trois termes suivante:

$$P_{k+1}(x) = (A_{k+1}x + B_{k+1})P_k(x) - C_{k+1}P_{k-1}(x), \quad (1.7)$$

avec pour conditions initiales $P_{-1} = 0$ et $P_0 = 1$.

Les coefficients A_{k+1} , B_{k+1} et C_{k+1} vérifient

$$A_{k+1} = -\frac{c(P_{k+1}^2)}{c(xP_kP_{k+1})}$$

$$B_{k+1} = -\frac{c(P_{k+1}^2)c(xP_k^2)}{c(xP_kP_{k+1})c(P_k^2)}$$

et

$$C_{k+1} = \frac{c(P_{k+1}^2)c(xP_{k-1}P_k)}{c(xP_kP_{k+1})c(P_{k-1}^2)}$$

La récurrence ne permet pas, sous cette forme, d'obtenir les P_i car les coefficients dépendent du polynôme à calculer. Par contre, dès qu'une condition de normalisation est imposée, les coefficients sont déterminés. Dans le cas où les P_i sont unitaires, la relation de récurrence devient

$$P_{k+1}(x) = (x + \alpha_{k+1})P_k(x) + \beta_{k+1}P_{k-1}(x) \quad (1.8)$$

avec pour conditions initiales $P_{-1} = 0$ et $P_0 = 1$. Les coefficients α_{k+1} et β_{k+1} sont calculés de la manière suivante:

$$\beta_{k+1} = -\frac{c(x^k P_k)}{c(x^{k-1} P_{k-1})}$$

et

$$\alpha_{k+1} = \frac{c(x^k P_{k-1})}{c(x^{k-1} P_{k-1})} - \frac{c(x^{k+1} P_k)}{c(x^k P_k)}$$

Différentes conditions de normalisation peuvent être utilisées pour le calcul des polynômes orthogonaux, chacune d'elles répondant à une problématique particulière. Des exemples de choix de normalisations sont étudiés dans [25].

1.3 Un problème modèle

Nous nous proposons d'observer un exemple d'application des polynômes orthogonaux. Nous ne détaillons pas tous les principes mathématiques. Mais nous en donnons la trame quitte à anticiper sur notre exposé. Nous avons choisi d'exposer au travers des approximants de Padé la variété des formulations et des algorithmes sans insister pour l'instant sur les problèmes auxquels ils donnent naissance. Le contexte est donc celui de l'approximation rationnelle du développement de Taylor. Nous illustrons par des exemples d'approximations d'une fraction par une autre fraction dont les numérateurs et dénominateurs sont de degrés plus petits.

Etant donné une fonction complexe f de la variable t et deux entiers p et q , on se propose de trouver, parmi les fonctions rationnelles qui ont un numérateur de degré

inférieur ou égal à p et un dénominateur de degré inférieur ou égal à q , celles dont le développement de Taylor en l'abscisse $t = 0$ coïncide avec celui de f le plus loin possible. La première méthode que nous rappelons ([15],[2]) repose sur le calcul des polynômes orthogonaux formels. Lorsque f est une fonction rationnelle, un autre procédé, basé sur les fractions continues ([78]), nous permet d'effectuer une comparaison numérique.

1.3.1 Les approximants de Padé

Partant d'une série formelle $f(t) = \sum_{i=0}^{\infty} c_i t^i$ on construit une fonction rationnelle $R(t)$, dont le numérateur $P(t)$ est de degré p exactement et le dénominateur $Q(t)$ est de degré q exactement, de telle sorte que le développement de $R(t)$ suivant les puissances croissantes de la variable t coïncide avec celui de $f(t)$ jusqu'au terme où l'exposant a la plus grande valeur possible ([15]). Les choix de $P(t)$ et $Q(t)$ nous laissent à priori $p + q + 2$ degrés de liberté correspondant aux coefficients. Une fraction n'est cependant unique qu'à un coefficient de normalisation près. On ne peut donc espérer reconstituer au maximum que $p + q + 1$ termes de la série $f(t)$. En posant cette contrainte comme point de départ, on a

$$f(t) - \frac{P(t)}{Q(t)} = O(t^{p+q+1}). \quad (1.9)$$

Les $p + q + 1$ termes qui se neutralisent concernent les degrés 0 à $p + q$. En multipliant par $Q(t)$ l'équation prend également la forme suivante

$$Q(t)f(t) - P(t) = O(t^{p+q+1}). \quad (1.10)$$

Les égalités (1.9) et (1.10) ne sont pas équivalentes à cause des simplifications éventuelles des degrés ([2], p.20). On note $P(t) = \sum_{i=0}^p a_i t^i$ et $Q(t) = \sum_{i=0}^q b_i t^i$. On développe l'expression (1.10) puis on identifie les coefficients termes après termes. Afin de ne pas différencier les cas de figure $p \leq q$ et $p > q$ on adopte pour convention que $c_i = 0$ pour tout $i < 0$. Pour les $p + 1$ premiers termes cela permet d'obtenir les conditions suivantes

$$\begin{cases} c_0 b_0 & = a_0 \\ c_1 b_0 + c_0 b_1 & = a_1 \\ \dots & \dots \dots \\ c_p b_0 + c_{p-1} b_1 + \dots + c_{p-q} b_q & = a_p \end{cases} \quad (1.11)$$

Ensuite, les q termes suivants fournissent des conditions qui ne portent plus que sur les coefficients b_i .

$$\begin{cases} c_{p+1} b_0 + c_p b_1 + \dots + c_{p-q+1} b_q & = 0 \\ c_{p+2} b_0 + c_{p+1} b_1 + \dots + c_{p-q+2} b_q & = 0 \\ \dots & \dots \dots \\ c_{p+q} b_0 + c_{p+q-1} b_1 + \dots + c_p b_q & = 0 \end{cases} \quad (1.12)$$

1.3.2 Méthode s'appuyant sur les polynômes orthogonaux formels

Le dernier système linéaire n'est autre qu'un système de Hankel définissant des polynômes orthogonaux formels non normalisés. Il est donc naturel d'imposer la condition de normalisation $b_0 = 1$ sur $Q(t)$, tant pour choisir une normalisation pour la fraction que pour distinguer la fonctionnelle qui définit l'orthogonalité. En écrivant le système (1.12) sous la forme

$$\begin{pmatrix} c_{p-q+1} & c_{p-q+2} & \cdots & c_{p-1} & c_p \\ c_{p-q+2} & c_{p-q+3} & \cdots & c_p & c_{p+1} \\ \vdots & \vdots & & \vdots & \vdots \\ c_{p-1} & c_p & \cdots & c_{p+q-3} & c_{p+q-2} \\ c_p & c_{p+1} & \cdots & c_{p+q-2} & c_{p+q-1} \end{pmatrix} \begin{pmatrix} b_q \\ b_{q-1} \\ \vdots \\ b_2 \\ b_1 \end{pmatrix} = - \begin{pmatrix} c_{p+1} \\ c_{p+2} \\ \vdots \\ c_{p+q-1} \\ c_{p+q} \end{pmatrix}, \quad (1.13)$$

on remarque que les coefficients $1, b_1, b_2, \dots, b_q$ apparaissent comme les coefficients renversés du polynôme $S_q(x)$ de degré q unitaire et orthogonal par rapport à la fonctionnelle c de moments $(c_{p-q+1}, c_{p-q+2}, \dots, c_{p+q-1}, c_{p+q})$. $Q(x)$ est le polynôme réciproque de $S_q(x)$, c'est-à-dire $Q(x) = x^q S(1/x)$. Le numérateur de l'approximant vérifie quant à lui un système linéaire qui le définit comme le polynôme réciproque du polynôme associé (3.3) à $S_q(x)$ par la fonctionnelle c .

$$\begin{pmatrix} c_0 & c_1 & \cdots & c_{p-1} & c_p \\ 0 & c_0 & \cdots & c_{p-2} & c_{p-1} \\ \vdots & 0 & & \vdots & \vdots \\ 0 & \vdots & & c_0 & c_1 \\ 0 & 0 & \cdots & 0 & c_0 \end{pmatrix} \begin{pmatrix} b_0 \\ b_1 \\ \vdots \\ \vdots \\ b_q \end{pmatrix} = \begin{pmatrix} a_p \\ a_{p-1} \\ \vdots \\ \vdots \\ a_0 \end{pmatrix}. \quad (1.14)$$

Les fractions que l'on a ainsi construites sont des approximants de Padé que l'on note $[p/q]_f$ en référence aux degrés des numérateurs et dénominateurs. Lorsqu'il n'y a pas d'ambiguïté sur la fonction f de référence on peut omettre l'indice. Les systèmes (1.12) et (1.13) ne sont pas équivalents. Contrairement au système (1.12) qui admet toujours une solution non triviale d'après un théorème de Frobenius ([55], ch. 5), le système (1.13) peut ne pas avoir de solution. Cette distinction permet de clarifier les nuances entre les définitions de forme rationnelle de Padé, de forme rationnelle réduite de Padé et d'approximants de Padé. Les formes rationnelles de Padé et leurs réduites existent toujours, la forme réduite permettant d'obtenir l'unicité. Par contre, certaines formes réduites ne sont plus des formes de Padé, c'est pourquoi en introduisant une condition de normalisation on obtient une définition qui conserve le critère d'unicité au détriment du critère d'existence. Ces précisions sont détaillées dans le livre de J. Gilewicz ([55], ch. 5) consacré aux approximants de Padé.

De cette présentation, se déduit une méthode de calcul des approximants de Padé basée sur les polynômes orthogonaux formels ([15]). Dans certains problèmes pratiques, les moments peuvent être obtenus comme résultats numériques provenant de mesures

physiques. Ils peuvent découler d'une intégration lorsque la fonctionnelle est donnée par une mesure définie par une intégrale. Nous supposons donc connus les moments de la fonctionnelle. La récurrence à trois termes et ses extensions ([39]) permettent de calculer les polynômes orthogonaux formels relativement à la fonctionnelle donnée. Pour résoudre les systèmes (1.13) et obtenir les dénominateurs, il suffit de renverser les coefficients des polynômes orthogonaux obtenus par la récurrence à trois termes. En calculant les associés des polynômes orthogonaux puis en les renversant, on trouve les numérateurs. Les approximants de Padé obtenus sont alors de la forme $[k/(k+1)]_f$.

Exemple 2

Prenons une fraction de la forme $f(x) = p(x)/(x^6 - 1)$, avec

$$p(x) = 1 + 2x + x^2 - 2x^3 + x^4 - 2x^5 + x^6.$$

Du développement de Taylor de f on extrait les moments de la fonctionnelle associée.

$$f(x) = -1 - 2x - x^2 + 2x^3 - x^4 + 2x^5 - 2x^6 - 2x^7 - x^8 + 2x^9 + O(x^{10}).$$

Les moments sont donc donnés par $c = \{-1, -2, -1, 2, -1, 2, -2, -2, -1, 2, \dots\}$. En appliquant la récurrence à trois termes, on obtient les polynômes orthogonaux par rapport à c ,

$$\begin{aligned} p_0(x) &= 1, \\ p_1(x) &= x - 2, \\ p_2(x) &= x^2 - \frac{4x}{3} + 5/3, \\ p_3(x) &= x^3 + \frac{5x^2}{4} + 3/4, \\ p_4(x) &= x^4 + \frac{21x^3}{8} + \frac{x}{4} + \frac{67x^2}{32} + \frac{53}{32}, \end{aligned}$$

et avec la même récurrence initialisée différemment on trouve les associés (3.3),

$$\begin{aligned} q_0(x) &= 0, \\ q_1(x) &= -1, \\ q_2(x) &= -x - 2/3, \\ q_3(x) &= -x^2 - \frac{13x}{4} - 7/2, \\ q_4(x) &= -x^3 - \frac{37x^2}{8} - \frac{267x}{32} - \frac{81}{16}, \end{aligned}$$

où q_i de degré $i - 1$ est l'associé de p_i . Il ne reste plus qu'à renverser les coefficients des polynômes pour pouvoir former un approximant de Padé $[n/(n+1)]$, par exemple

$$x^2 p_2\left(\frac{1}{x}\right) = \frac{5x^2}{3} - \frac{4x}{3} + 1,$$

$$xq_1\left(\frac{1}{x}\right) = -\frac{2x}{3} - 1$$

et donc

$$[1/2]_f = \frac{-2x - 3}{5x^2 - 4x + 3}.$$

◇

Appliquons ce procédé aux fonctions définies implicitement par une équation algébrique de la forme

$$P(x, y) = 0,$$

où P_n est un polynôme en les variables x et y , les solutions étant cherchées sous la forme d'une fonction d'une variable réelle $y = f(x)$.

Exemple 3

Etudions les approximants de Padé que l'on peut construire à partir de l'équation algébrique

$$y^2 + 2xy + 1 = 0.$$

Les solutions exactes, obtenues en résolvant l'équation du second degré, $f_1(x) = -x + \sqrt{x^2 - 1}$ et $f_2(x) = -x - \sqrt{x^2 - 1}$, ne sont pas définies pour $x \in]-1, 1[$. En se restreignant localement en $x = 5/3$, on peut se ramener, par changement de variable, à une étude locale en $x = 0$ de l'équation algébrique

$$y^2 + 2xy + \frac{10y}{3} + 1 = 0 \tag{1.15}$$

En regard de la méthode précédente il suffit d'obtenir un développement de Taylor en $x = 0$ pour reconstituer un approximant rationnel en $x = 0$. On pose $y(x) = a_0 + a_1x + a_2x^2 + \dots + O(x^n)$ que l'on substitue à y dans (1.15). Les coefficients a_i sont déterminés successivement de telle sorte que l'on augmente la valuation (ie le plus bas degré) du développement obtenu. A partir du développement

$$a_0^2 + \frac{10a_0}{3} + 1 + \left(\frac{10a_1}{3} + 2a_0a_1 + 2a_0\right)x + \dots = 0,$$

commençons par éliminer le coefficient constant. La contrainte obtenue est

$$a^2 + \frac{10a}{3} + 1 = 0.$$

Les deux solutions possibles pour a_0 décrivent le coefficient constant du développement de chacune des deux branches de l'équation algébrique. Une fois choisi, le nombre a_0 détermine la fonction algébrique que l'on développe. Les coefficients a_i sont alors

déterminés de manière unique à partir de ceux qui les précèdent. Dans notre cas nous trouvons au voisinage de $x = 5/3$,

$$f_1(x) = -1/3 + \frac{(x - \frac{5}{3})}{4} - \frac{27(x - \frac{5}{3})^2}{128} + \frac{405(x - \frac{5}{3})^3}{2048} - \frac{26487(x - \frac{5}{3})^4}{131072} + O((x - \frac{5}{3})^5)$$

et

$$f_2(x) = -3 - \frac{9(x - \frac{5}{3})}{4} + \frac{27(x - \frac{5}{3})^2}{128} - \frac{405(x - \frac{5}{3})^3}{2048} + \frac{26487(x - \frac{5}{3})^4}{131072} + O((x - \frac{5}{3})^5).$$

On peut alors extraire des approximants de Padé. Par exemple, avec le développement qui précède

$$[1/2]_{f_2} = \frac{-24 + 40x}{-7 - 6x + x^2}.$$

On peut donc construire facilement un approximant de Padé d'une solution d'une équation algébrique, ce qui n'est pas toujours le cas pour obtenir une solution formelle. Pour une fonction localement holomorphe en x_0 , l'unicité du développement de Taylor en x_0 assure l'unicité des approximants de Padé réguliers d'un ordre donné. Pour chacune des branches, f_1 et f_2 étant holomorphes au voisinage de $x = 5/3$, les approximants de Padé obtenus par la relation implicite coïncident avec les approximants de Padé des solutions exactes. Observons sur notre exemple l'erreur que commettent les approximants de Padé de f_2 à différents ordres au voisinage de $x = 5/3$. Le graphique (1) illustre bien la qualité de l'approximation dès les premiers approximants. Nous comparons également (2) le développement de Taylor tronqué avec un approximant de Padé utilisant le même nombre de moments.

◇

Il est clair que l'approximant de Padé parvient à extraire plus d'informations de la série formelle. Tronquer une série est une opération beaucoup plus réductrice qu'il n'y paraît. Il faut bien penser qu'une série entière possède suffisamment d'informations pour calculer la valeur de la fonction n'importe où sur le domaine de convergence et même au-delà par le principe de re-sommation de Weierstrass. Or justement, en changeant la forme de la variable par rapport à laquelle on développe, on fait intervenir l'infinité des moments. Cesser cette infinité fait perdre les propriétés induites par la re-sommation. Faire de la re-sommation avec une forme tronquée (un polynôme) n'est pas raisonnable puisque l'on néglige déjà toute l'information sur la disposition des pôles. L'approximation polynômiale reste bonne tant que le domaine d'approximation est éloigné de tout pôle. Au contraire, les pôles à proximité du domaine d'approximation peuvent être neutralisés par une approximation rationnelle à condition de faire intervenir suffisamment de puissances de x , sauf pour les singularités essentielles. Les approximants de Padé prolongent aux fractions l'idée de l'interpolation polynomiale (de Hermite), l'approximation de Padé est donc optimale en ce sens (localisation des pôles) et cela la rend plus adaptée à l'application approximative du principe de re-sommation. Plus précisément, les approximants de Padé

apparaissent, d'après un théorème de Walsh ([2], p.662), comme les meilleures approximations locales rationnelles. Posons-nous maintenant le problème inverse, c'est-à-dire retrouver l'équation algébrique à partir d'approximants de Padé. Pour aborder ce point de vue, nous présentons une deuxième interprétation des approximants de Padé.

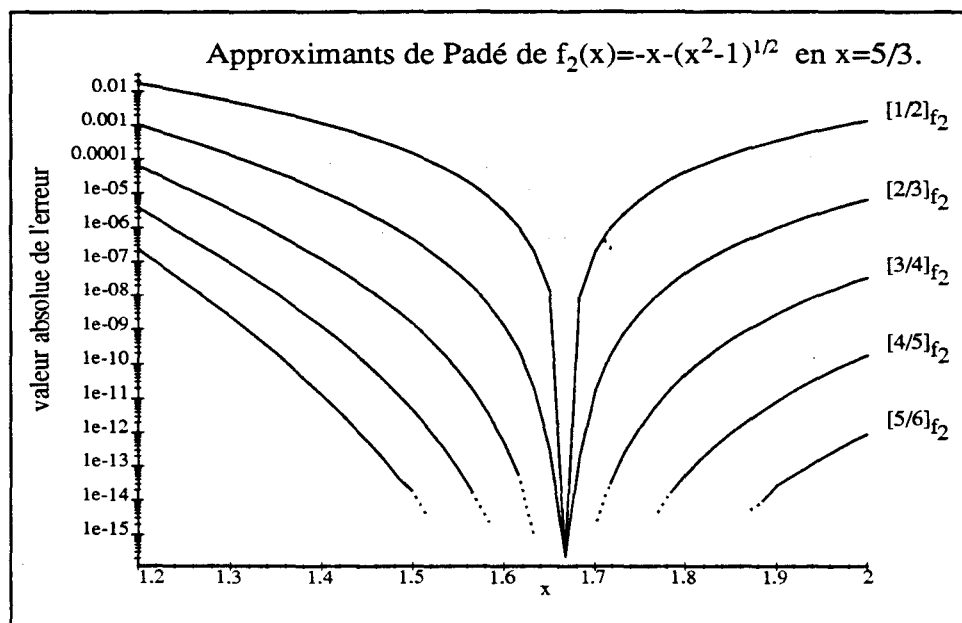


fig. 1

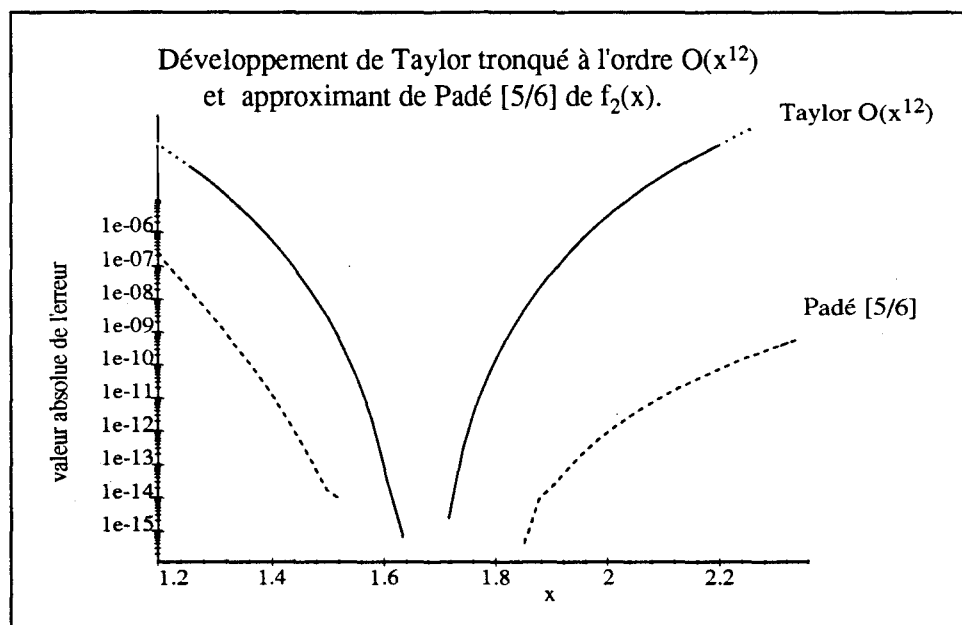


fig. 2

1.3.3 Méthode reposant sur les fractions continues

Lorsque l'on cherche un approximant de Padé d'une fonction f , on ne dispose pas nécessairement du développement de Taylor de la fonction. Prenons le cas où f est une fonction rationnelle. Le développement de Taylor peut alors s'obtenir par division suivant les puissances croissantes de la variable. Cette division peut cependant servir de manière détournée à construire un développement en fraction continue dont les convergents sont eux-mêmes des approximants de Padé ([2]).

Définition 1 (Les C-fractions) Une C-fraction est une fraction continue de la forme

$$f(x) = a_0 + \frac{a_1 x^{\alpha_1}}{1 + \frac{a_2 x^{\alpha_2}}{1 + \frac{a_3 x^{\alpha_3}}{1 + \dots}}}$$

où $a_i \in \mathbb{C}$ et $\alpha_i \in \mathbb{N}$. La C-fraction tronquée à l'ordre k définit le convergent d'ordre k .

Les C-fractions ont la propriété d'être en correspondance bi-univoque avec l'espace des séries de Taylor ([78], th. 5 p. 253). La construction de cette correspondance entre une série S et un C-fraction T s'établit sur le fait que le n -ième convergent de T doit avoir un développement de Taylor qui coïncide à l'ordre n avec celui de S . En particulier, une C-fraction qui se termine correspond au développement d'une fraction rationnelle. Lorsque $\alpha_i = 1$ pour tout i , les C-fractions sont dites régulières. Le développement en C-fraction de la série $S(z) = \sum_{k=0}^{\infty} c_k z^k$ est régulier et infini si et seulement si ([78], th. 6 p. 257) $H_k^{(1)} H_k^{(2)} \neq 0$ pour $k = 1, 2, \dots$, avec

$$H_k^{(i)} = \begin{vmatrix} c_i & c_{i+1} & \cdots & c_{i+k-1} \\ c_{i+1} & c_{i+2} & \cdots & c_n \\ \vdots & \vdots & & \vdots \\ c_{i+k-1} & c_{i+k} & \cdots & c_{i+2k-2} \end{vmatrix}$$

Dans cette situation, le convergent d'ordre $2k$ est une fraction dont le numérateur est de degré k , le dénominateur degré k et dont le développement de Taylor coïncide à l'ordre $2k$ avec celui de la série S . Par unicité des approximants de Padé on reconnaît la propriété de $[k/k]_S$, donc le convergent donne un approximant de Padé. A l'ordre $2k + 1$ le convergent obtenu est $[(k + 1)/k]_S$. L'algorithme de calcul des approximants qui en découle est celui de Viscovatov ([2], p.133) et sa forme généralisée ([32]). Donnons un exemple de construction du développement en C-fraction.

Exemple 4

Développons en C-fraction continue la fraction suivante,

$$f(x) = \frac{1 - 2x + x^2}{1 + x}$$

$$\begin{aligned}
f(x) &= 1 + \frac{-3x + x^2}{1+x} \\
&= 1 - \frac{3x}{1+x} \\
&= \frac{1 - \frac{1}{3}x^2}{1 - \frac{1}{3}x^2} \\
&= 1 - \frac{3x}{1+x \frac{1 + \frac{1}{3}x}{-3x + x^2}} \\
&= 1 - \frac{3x}{1 + \frac{x}{1 - \frac{1}{3}x \frac{1+x}{1 + \frac{1}{3}x}}} \\
&= 1 - \frac{3x}{1 + \frac{x}{1 - \frac{\frac{1}{3}x}{1 - \frac{\frac{2}{3}x}{1+x}}}}
\end{aligned}$$

◇

En application, on peut retrouver maintenant une équation algébrique à partir des approximants de Padé d'une fonction algébrique dont le développement en C-fraction est régulier et périodique.

Exemple 5

Nous allons reconstituer une équation algébrique vérifiée par $f_2(x) = -x + \sqrt{x^2 - 1}$ et cela uniquement à l'aide de son développement en C-fraction continue en $x = 5/3$.

$$f_2(x) = -\frac{1}{3} + \frac{\frac{1}{4}x}{1 + \frac{\frac{27}{32}x}{1 + \frac{\frac{3}{32}x}{1 + \frac{\frac{27}{32}x}{1 + \frac{\frac{3}{32}x}{1 + \dots}}}}}$$

Le développement en C-fraction obtenu est régulier et périodique. On pose

$$z = 1 + \frac{\frac{27}{32}x}{1 + \frac{\frac{3}{32}x}{1 + \frac{\frac{27}{32}x}{1 + \frac{\frac{3}{32}x}{1 + \dots}}}$$

Alors z vérifie la relation suivante,

$$z = 1 + \frac{\frac{27}{32}x}{1 + \frac{\frac{3}{32}x}{z}},$$

z satisfait donc à l'équation algébrique,

$$32z^2 + (-24x - 32)z - 3x = 0.$$

De plus, y et z vérifient

$$y = -1/3 + \frac{x}{4z}$$

dont on peut extraire z ,

$$z = \frac{3x}{12y + 4}.$$

En reportant dans l'équation algébrique de z et après simplifications, on retrouve une équation algébrique satisfaite par y ,

$$y^2 + (2x + 10/3)y + 1 = 0.$$

Nous retrouvons l'équation algébrique (1.15) sans avoir utilisé la forme explicite de la fonction.

◇

Remarquons enfin que la recherche des approximants de Padé consiste à construire une équation algébrique de degré 1 dont la fonction y donnée doit être une solution approchée,

$$F(x, y) = P(x)y - Q(x) = O(x^k).$$

Le degré d'approximation k de l'équation algébrique est lié aux degrés de libertés introduits par les degrés de $P(x)$ et $Q(x)$. Le développement de Taylor est le cas particulier où l'on impose $P(x) = 1$. Cela permet de concevoir une généralisation de l'approximation de Padé, à savoir, trouver une équation algébrique de degré n donné dont la fonction y donnée soit une solution approchée à l'ordre k .

$$P_n(x)y^n + P_{n-1}(x)y^{n-1} + \dots + P_0(x) = O(x^k).$$

On obtient ainsi un cas particulier d'approximants de Hermite-Padé ([2], p.495). k et n sont encore reliés par les degrés de libertés du problème. La solution \tilde{y} , de l'équation algébrique précédente sans second membre, réalise ainsi une approximation algébrique de la fonction y . Les approximants de Padé apparaissent comme un cas particulier d'approximation d'ordre un, au même titre qu'une approximation affine pour le développement de Taylor. Cet ordre ne reflète pour l'instant que le degré de l'équation algébrique. En l'abscisse où l'équation algébrique représente la série formelle jusqu'à l'ordre k , si nous

calculons un approximant de Padé de \tilde{y} qui ne fait pas intervenir plus de k termes alors ce sera également un approximant de Padé de y . Par contre un procédé de re-sommation appliqué à l'approximant de Padé ne donnera pas le même résultat que le changement de variable correspondant appliqué à l'approximation algébrique. L'information apportée par une approximation algébrique contient au moins celle contenue dans l'approximation de Padé. Néanmoins si elle est obtenue à partir d'un approximant de Padé, elle n'en contiendra pas davantage. Il reste à savoir si on peut construire de tels approximants algébriques par un autre biais (par des approximants de Padé en deux points?) et quel niveau d'information cela pourrait conserver au travers du procédé de re-sommation. Quoiqu'il en soit, pour en revenir au calcul des valeurs locales, l'approximation de Padé est la plus indiquée.

1.3.4 Une comparaison numérique

Dans les deux méthodes de calculs d'approximants, les suites de coefficients numériques sur lesquels reposent les récurrences ne sont pas les mêmes. Alors que la première méthode se base sur les coefficients de la fonctionnelle, la deuxième méthode utilise comme intermédiaires les coefficients des numérateurs partiels de la fraction continue. Il est intéressant de comparer les comportements numériques de ces algorithmes lorsqu'il s'agit de calculer des approximants de Padé d'une fraction donnée. Même si ce n'est pas systématique, les graphiques obtenus montrent régulièrement un avantage pour le calcul à partir des coefficients de la récurrence à trois termes. Cependant, aucune des deux méthodes que nous avons programmées pour la comparaison n'utilise de procédé de renormalisation, ni de saut, ni même de re-sommation. Pour chaque fonctionnelle, nous avons calculé un approximant exact en des abscisses rationnelles. En arithmétique à virgule flottante de 16 chiffres de précision nous avons calculé un approximant par la relation à trois termes que l'on a évalué sur les valeurs numériques des abscisses (méthode directe), puis on a procédé à une autre évaluation en ces mêmes abscisses en utilisant seulement les coefficients de la récurrence et enfin, nous avons calculé numériquement les coefficients de la fraction continue dont on s'est servi pour évaluer récursivement les valeurs numériques des numérateurs et dénominateurs des convergents.

Exemple 6

Nous avons sélectionné deux exemples pour lesquels la comparaison est claire, mais ce n'est pas toujours le cas. Pour le premier exemple (fig. 3) nous avons pris la fonctionnelle dont les premiers moments sont donnés par

$$c = (1, 1, 5, 1, -2, -1, 8, 10, 10, -2, 1, -1, 1, 5, 10, -10, -11, 12, \dots).$$

L'exemple suivant (fig. 4), avec

$$c = (1/2, 1/2, 3/8, 2/3, 5/2, 3/2, 7/8, 4/3, 9/2, 5/2, 11/8, 2, 13/2, 7/2, 15/8, 8/3, 17/2, \dots),$$

montre que l'avantage ne revient pas toujours à la même méthode.

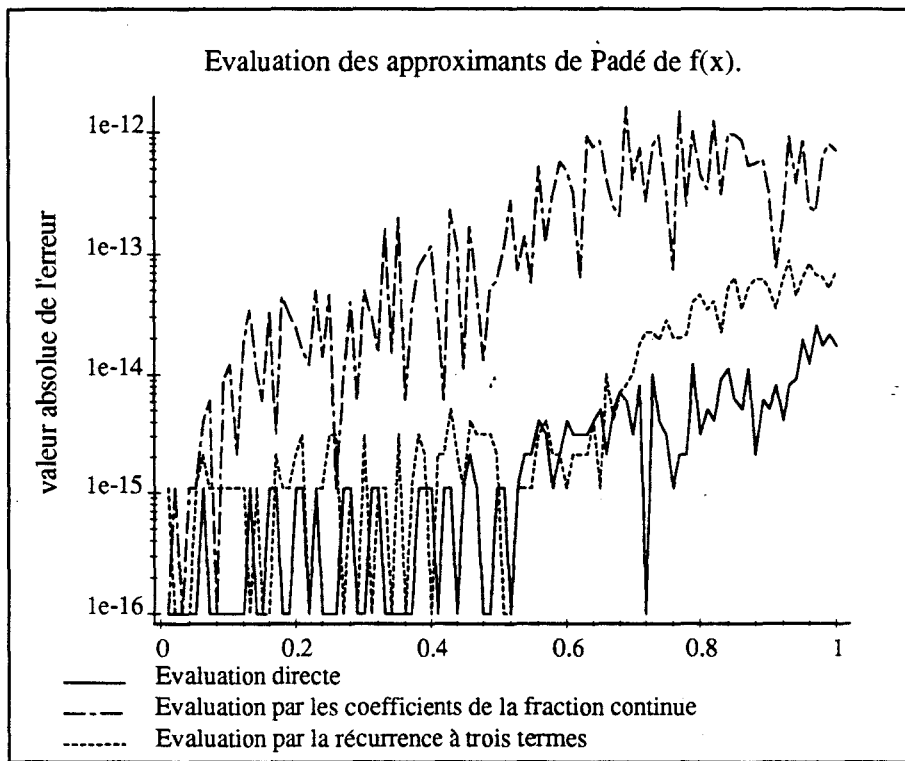


fig. 3

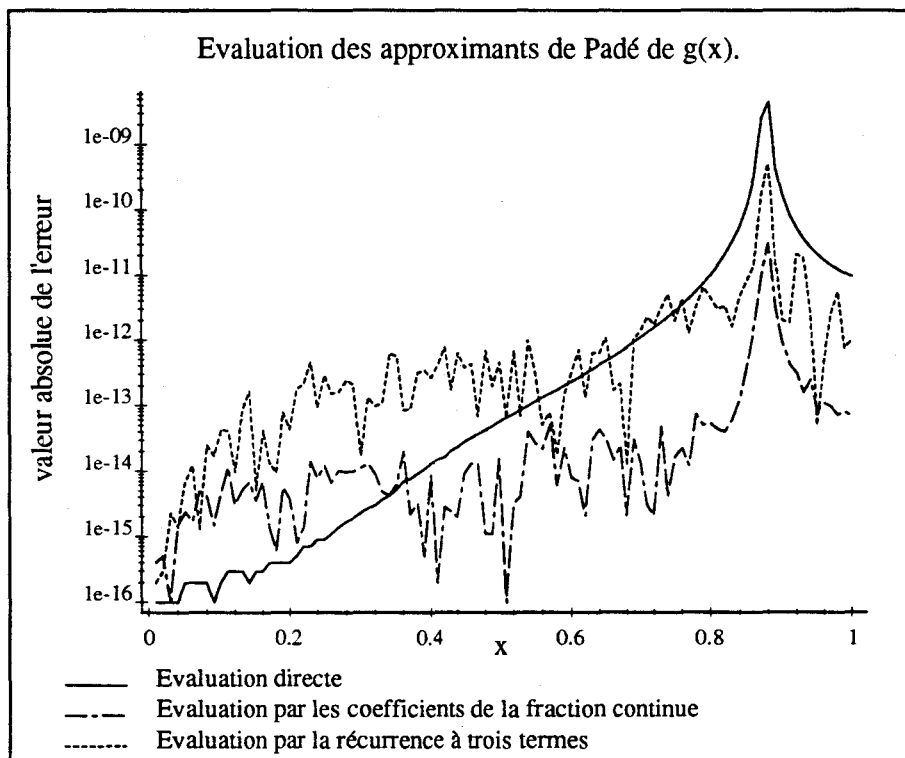


fig. 4

1.3.5 Perspectives de recherches

Nous avons présenté des points de vue numériques et formels sans les situer dans les perspectives de recherches actuelles. Reformulons notre problème algébrique pour en saisir tout l'intérêt. Le développement en fraction continue d'une fonction irrationnelle algébrique n'est pas fini, contrairement au cas des fonctions rationnelles. Une racine réelle d'une équation algébrique possède les mêmes propriétés et cela permet de représenter un nombre algébrique irrationnel par une équation algébrique ([37]). On peut alors faire des calculs sur ordinateur dans l'extension algébrique qu'elle engendre. On peut imaginer un sort identique pour les fonctions algébriques. Toute la difficulté réside dans la représentation. Est-elle unique? Comment reconnaître l'équation algébrique à partir du développement en fractions continues? Regroupons quelques problèmes que l'on peut associer intuitivement. Le développement en fractions continues d'un nombre irrationnel algébrique donne des approximations optimales ([78]) par le biais des convergents. Un théorème de Guelfond ([62]) précise, en sens inverse, que l'erreur d'approximation commise par le k -ième convergent est de la forme

$$\left| x - \frac{p}{q} \right| > \frac{\alpha}{q^k}$$

où α est un nombre qui ne dépend pas de x . Autrement dit, les approximations successives ne peuvent pas s'approcher très près d'un nombre irrationnel algébrique. Qu'en est-il des fonctions algébriques? Il existe bien sûr des théorèmes de convergences pour les approximations de Padé, comme le théorème de Montessus de Ballore ([16]), mais dans le cas d'abscisses rationnelles, ne peut-on pas obtenir un majorant de la vitesse de convergence? L'algorithme d'Euclide programmé en arithmétique exacte par la méthode des sous-résultants permet de maintenir une croissance linéaire des coefficients ([28]) or il a une base commune avec l'algorithme de développement en fractions continues. Peut-on obtenir un résultat similaire pour le calcul des polynômes orthogonaux mené en arithmétique exacte? Enfin, citons un résultat empirique sur la précision du calcul des approximations de Padé ([2], p.42). «... *approximately* M decimal places of accuracy are lost in the calculation of an $[L/M]$ approximant by direct solution of the (Hankel) linear system.>>. Y-a-t-il un lien avec la croissance linéaire des coefficients pour la méthode des sous-résultants? Les difficultés de calculs dépendent du choix de l'arithmétique utilisée. Nous voulons souligner que là où l'arithmétique exacte est confrontée à l'accroissement de la taille des coefficients, les arithmétiques numériques de précision finie conduisent parallèlement à la propagation et l'accumulation d'erreurs. L'arithmétique exacte permet d'utiliser des méthodes de projection (p -adiques ou modulaires) pour décomposer le problème de la croissance des coefficients. Quelles méthodes de projection peut-on envisager en arithmétique de précision finie? Les deux aspects formels et numériques ont des applications importantes et requièrent des moyens de contrôler leurs défauts.

1.4 Panorama des applications

Nous proposons à présent un tour d'horizon des domaines qui sont reliés aux polynômes orthogonaux. Après avoir résumé la situation des codes correcteurs, qui représente un cas typique d'application utilisant l'arithmétique des corps finis, nous décrivons brièvement une situation numérique extraite du traitement du signal, où l'on emploie essentiellement une arithmétique de précision finie. Enfin, nous ébauchons une synthèse de liens et de références qui nous guident parmi de nombreux domaines qui s'associent aux polynômes orthogonaux et nous l'accompagnons d'un schéma synoptique.

1.4.1 Codes correcteurs d'erreurs

Les systèmes de Hankel apparaissent au cours de la résolution du problème de décodage de certains codes correcteurs d'erreurs. Dans cette situation, le corps de base est fini \mathbb{F}_q où q est une puissance d'un nombre premier. Nous résumons aussi brièvement que possible le contexte de cette application. Pour toutes les définitions et notions d'algèbre utilisées, nous nous référons à ([89]).

Le codage et le transfert de l'information sont en général perturbés par de nombreux parasitages que l'on ne peut contourner. Les répétitions dans les conversations courantes sont certainement une manière spontanée d'éviter les malentendus de la communication. Une solution pour permettre la correction est donc la redondance, comme par exemple, l'utilisation dans certaines arithmétiques machines d'un bit de parité pour révéler la présence d'une erreur dans les calculs élémentaires. Pour détecter une erreur, il faut que le mot reçu ne fasse plus partie du dictionnaire utilisé, comme un changement de parité. Pour corriger l'erreur, il faut que l'orthographe du mot reçu ne soit proche que d'un seul mot du dictionnaire. Pour donner un sens mathématique au codage, on orthographie les mots avec les éléments d'un corps fini, en général \mathbb{F}_q . Pour simplifier les transferts et la détection des mots on travaille avec des mots de longueur fixe n , donc dans un espace vectoriel \mathbb{F}_q^n . On convient à l'avance du choix des vecteurs significatifs (les mots du code) de manière à les rendre aussi distants que possible pour la distance de Hamming (i.e. qui mesure la distance entre leurs orthographes). On impose également que l'ensemble des vecteurs significatifs forme un sous espace-vectoriel de \mathbb{F}_q^n ; on parle alors de codes linéaires.

Dans le cas particulier des codes cycliques, toutes les permutations circulaires d'un mot du code sont des mots du code. Dans cette situation et après transformation dont on passe le détail, le code devient un idéal (principal) de l'anneau $A = \mathbb{F}_q[x]/(x^n - 1)$ et les mots du code sont tous les multiples d'un seul élément $g(x)$ qu'on appelle le générateur du code. L'idéal précédent est le noyau d'un morphisme d'anneaux surjectif de $\mathbb{F}_q[x]$ sur A , en sorte qu'il existe $h(x)$ tel que $x^n - 1 = g(x)h(x)$. Le reste de la division modulo $g(x)$ détermine la classe du mot reçu modulo les mots du code. La classe d'un mot, qu'on appelle aussi le syndrome, contient une information suffisante pour localiser et corriger un certain nombre d'erreurs. Les composantes du syndrome indiquent les projections du mot sur l'espace orthogonal aux mots du code. En particulier, un mot du code est caractérisé

par un syndrome dont toutes les composantes sont nulles.

Le calcul du syndrome polynôme d'un mot reçu $y(x) \in A$ est simplement le reste de la division euclidienne de $y(x)$ par $g(x)$, mais on peut aussi le caractériser par les valeurs de $y(x)$ sur les racines de $g(x)$. La première difficulté est d'extraire du syndrome la localisation des fautes d'orthographe dans le mot reçu.

Détaillons précisément cette étape qui repose sur un système de Hankel. On note L le corps de décomposition de $x^n - 1$ sur \mathbb{F}_q et α une racine primitive de L . On suppose que $g(x)$ est le produit des polynômes minimaux (sur \mathbb{F}_q) de $\alpha^r, \alpha^{r+1}, \dots, \alpha^{r+2s+1}$, des puissances consécutives de α . $g(x)$ possède pour seules racines les nombres précédents et leurs conjugués. Si un polynôme de A admet une racine α^i , alors il admet également toutes ses conjuguées pour racines. On contrôle donc les classes modulo $g(x)$ par l'évaluation des polynômes sur les seules puissances consécutives données ci-dessus. Le mot reçu $y(x) \in A$ provient d'un mot du code $c(x) \in (g(x))$ entaché d'une erreur $\epsilon(x) \in A$. De l'égalité

$$y(x) = c(x) + \epsilon(x), \quad (1.16)$$

on déduit que $y(x)$ et $\epsilon(x)$ sont dans la même classe modulo $g(x)$ et donc

$$y(\alpha^j) = \epsilon(\alpha^j) = \sum_{i=0}^{n-1} \epsilon_i \alpha^{ji} \quad \text{pour } j = 1, \dots, r + 2s + 1.$$

On note I l'ensemble des indices i pour lesquels il y a une erreur $\epsilon_i \neq 0$. Le syndrome S de $y(x)$ est représenté par les valeurs

$$S_j = \epsilon(\alpha^j) = \sum_{i \in I} \epsilon_i \alpha^{ji} \quad \text{pour } j = 1, \dots, r + 2s + 1. \quad (1.17)$$

Pour déterminer I on pose $f(x) = \prod_{i \in I} (1 - \alpha^i x)$. Connaissant $f(x)$ il suffira de l'évaluer sur les α^j pour $j = 1, \dots, r + 2s + 1$ afin de détecter les indices j donnant la valeur 0, et d'en déduire que les indices $n - j$ sont dans I . Le nombre d'erreurs corrigibles ne peut dépasser s pour des raisons de distance. Calculons les coefficients de $f(x) = 1 + f_1 x + \dots + f_k x^k$ avec $k \leq s$. Pour $i \in I$ et $r \leq j \leq r + 2s + 1$ on a les égalités

$$\epsilon_j (\alpha^i)^{j+k} f(\alpha^{-i}) = 0,$$

$$\epsilon_j ((\alpha^i)^{j+k} + f_1 (\alpha^i)^{j+k-1} + \dots + f_k (\alpha^i)^j) = 0.$$

En sommant ces égalités pour $i \in I$, on obtient

$$\sum_{i \in I} \epsilon_j \alpha^{ij+k} + f_1 \sum_{i \in I} \epsilon_j \alpha^{ij+k-1} + \dots + f_k \sum_{i \in I} \epsilon_j \alpha^{ij} = 0,$$

c'est-à-dire,

$$S_{j+k} + f_1 S_{j+k-1} + \dots + f_k S_j = 0.$$

En regroupant ces équations pour $j = r, \dots, r + k$ cela nous donne le système de Hankel suivant,

$$\begin{pmatrix} S_r & S_{r+1} & \cdots & S_{r+k-2} & S_{r+k-1} \\ S_{r+1} & S_{r+2} & \cdots & S_{r+k-1} & S_{r+k} \\ \vdots & \vdots & & \vdots & \vdots \\ S_{r+k-2} & S_{r+k-1} & \cdots & S_{r+2k-4} & S_{r+2k-3} \\ S_{r+k-1} & S_{r+k} & \cdots & S_{r+2k-3} & S_{r+2k-2} \end{pmatrix} \begin{pmatrix} f_k \\ f_{k-1} \\ \vdots \\ \vdots \\ f_1 \end{pmatrix} = - \begin{pmatrix} S_{r+k} \\ S_{r+k+1} \\ \vdots \\ \vdots \\ S_{r+2k-1} \end{pmatrix}. \quad (1.18)$$

Ce système, de taille au plus $2s+1$, doit être résolu pour chaque mot reçu. Les valeurs des erreurs ϵ_i se déduisent du système de Vandermonde (1.17) et le mot du code est reconnu au moyen de la correction (1.16). Nous venons de décrire le décodage des codes B.C.H. et parmi eux figurent les codes de Reed-Solomon. Ces derniers se distinguent par un système de Hankel (1.18) triangulaire pour lequel on peut appliquer une méthode de résolution par transformée de Fourier rapide (3.3). Si nous anticipons sur les résultats que nous avons obtenus, on distingue manifestement la même différence entre ces deux types de codes qu'entre les méthodes de Hankel-Padé et les méthodes de type Padé. Pour aller au-delà dans ces analogies entre le codage et les polynômes orthogonaux, nous citerons deux phrases extraites de ([89], p.314) concernant l'interprétation des codes B.C.H. en termes de géométrie algébrique. << ... les codes B.C.H. s'obtiennent par descente (intersection) de codes géométriques sur une droite projective.>> << ... les codes de Reed-Solomon sont obtenus sans opération de descente.>> Ceci nous laisse envisager une description des approximants de Padé en termes de géométrie algébrique et vient renforcer l'intérêt que l'on pourra porter aux exemples (3 et 5) en liaisons avec les courbes algébriques.

1.4.2 Systèmes de Toeplitz en traitement du signal

Les polynômes orthogonaux ont une grande importance en traitement numérique du signal. La théorie des polynômes orthogonaux sur le cercle unité est la formulation la plus courante dans ce contexte ([38]). On représente un signal discret par une suite de nombres réels $s = (s_i)_{i \in \mathbb{Z}}$. On ne considère ici que les signaux d'énergie finie, c'est-à-dire $s \in l^2$ ou encore $\sum_{i \in \mathbb{Z}} |s_i|^2 < \infty$. La fonction d'autocorrélation de s est la suite $r = (R_n)_{n \in \mathbb{Z}}$ où

$$R_n = \sum_{i \in \mathbb{Z}} s_{n+i} \bar{s}_i \quad \text{pour } n \in \mathbb{Z}.$$

Nous allons observer comment ces nombres permettent une reconstruction approximative du signal. Le principe de prédiction linéaire consiste à choisir un entier k puis à prédire une valeur du signal \tilde{s}_n au temps n à partir d'une combinaison linéaire de k observations antérieures du signal, $s_{n-1}, s_{n-2}, \dots, s_{n-k}$. Les coefficients p_i de la combinaison linéaire cherchée

$$\tilde{s}_n = - \sum_{i=1}^k p_i s_{n-i},$$

sont choisis afin de minimiser au sens des moindres carrés le signal de l'erreur $e_n = s_n - \hat{s}_n$. En notant p le vecteur de composantes $(p_i)_{1 \leq i \leq k}$, c_i les vecteurs de la base canonique de \mathbb{C}^k et S la matrice de dimension $\infty \times k$

$$S = \begin{pmatrix} \vdots & \vdots & \cdots & \vdots \\ s_{n-1} & s_{n-2} & \cdots & s_{n-k} \\ s_n & s_{n-1} & \cdots & s_{n-k+1} \\ \vdots & \vdots & \cdots & \vdots \end{pmatrix}$$

la suite $e = (e_n)_{n \in \mathbb{Z}}$ a pour norme l^2 ,

$$\|e\|^2 = \|s - \hat{s}\|^2 = \|s + Sp\|^2.$$

La minimisation est obtenue lorsque les dérivées partielles de $\|e\|^2$ suivant les coefficients p_i sont nulles. Pour cela, il faut imposer

$$c_i^T S^T Sp + c_i^T S^T s = 0 \quad \text{pour } i = 1, \dots, k.$$

En introduisant les composantes de la fonction d'autocorrélation, la relation devient

$$\sum_{n=1}^k R_{i-n} p_n = -R_i \quad \text{pour } i = 1, \dots, k.$$

Traduit sous forme de système linéaire, cela donne le système de Toeplitz

$$\begin{pmatrix} R_0 & R_{-1} & \cdots & R_{-k+1} \\ R_1 & R_0 & \cdots & R_{-k+2} \\ \vdots & & & \vdots \\ R_{k-1} & R_{k-2} & \cdots & R_0 \end{pmatrix} \begin{pmatrix} p_1 \\ p_2 \\ \vdots \\ p_k \end{pmatrix} = - \begin{pmatrix} R_1 \\ R_2 \\ \vdots \\ R_k \end{pmatrix}.$$

Le traitement du signal, comme la reconnaissance vocale ou le traitement de l'image, requiert de plus en plus de traitement en temps réel. Cela impose des temps de calculs limités et nécessite de disposer d'algorithmes rapides et stables. Nous venons de voir que les systèmes de Toeplitz interviennent de manière naturelle dans la résolution du problème des moindres carrés discrets très fréquent dans ce domaine. L'algorithme de Levinson ([76]) a été développé dans cet esprit et fournit un procédé stable et rapide dans le cas des matrices de Toeplitz définies positives. Des évolutions comme le 'split Levinson algorithm' ([38]) et des techniques telles que le 'look-ahead' ont permis d'optimiser et d'étendre cette méthode. Une autre stratégie initiée par G.Strang ([99]) et dont les évolutions récentes sont décrites dans ([82],[83]) consiste à proposer un préconditionnement circulant des matrices de Toeplitz qui accélère la convergence de l'algorithme du gradient conjugué préconditionné.

1.4.3 Schéma synoptique

Les systèmes de Hankel, comme nous avons pu l'observer, relient directement les approximants de Padé et les polynômes orthogonaux. Les fractions continues, dont certaines formes conduisent aux approximants de Padé, sont le résultat de différentes formes de divisions. La division euclidienne traduit la progression régressive le long de la diagonale principale de la table des polynômes orthogonaux ([39]). La division suivant les puissances croissantes décrit la progression en forme de marches d'escalier entre la diagonale principale et sa première sous-diagonale ([78], p.374). Cela se traduit par plusieurs algorithmes possibles pour l'évaluation des approximants de Padé en une abscisse donnée. Outre la possibilité économique d'évaluer une fois pour toutes les coefficients de la fraction pour ensuite l'évaluer en chaque abscisse, on peut procéder à une récurrence du type ϵ -algorithme à partir des valeurs de la série formelle tronquée et que l'on renouvelle pour chaque abscisse. Il est également possible de calculer les coefficients de la récurrence puis d'appliquer un schéma de Hörner pour chaque abscisse, ou enfin de construire auparavant le développement en fractions continues et appliquer un des algorithmes d'évaluation de fractions continues présentés dans ([80]). Par exemple, dans le cas d'un développement en fractions continues, le qd -algorithme fournit les coefficients du développement et un double schéma de Hörner, pour les numérateurs puis les dénominateurs, permet ensuite l'évaluation pour chaque abscisse en une division. L' ϵ -algorithme ne nécessite pas de connaître l'approximant de Padé, en cela il établit une liaison avec les méthodes d'accélération de la convergence. Remarquons que les deux formes de divisions employées traduisent, au niveau des nombres entiers, les représentations modulaire et p -adique. La première décrit chaque nombre par ses projections sur les corps \mathbb{F}_p avec p premier, la seconde construit un développement asymptotique dans les \mathbb{F}_q où les entiers q sont des puissances croissantes d'un même nombre premier. La description p -adique tronquée de Krishnamurthy ([15], p.30) offre une interprétation en termes d'approximants de type Padé. La complémentarité des deux descriptions se retrouve également en théorie des codes correcteurs où l'opération de codage est obtenue par une division euclidienne par un polynôme irréductible sur un corps fini alors que l'opération de décodage requiert la résolution d'un système de Hankel. En remontant au niveau matriciel, cela se résume par le fait qu'une matrice de Hankel admet pour inverse une matrice Bezoutienne ([67], p.1063). Parallèlement, les systèmes de Toeplitz permettent d'élaborer une notion d'orthogonalité sur le cercle unité qui possède des propriétés similaires (récurrence, relation de Christoffel-Darboux...) ([31]). C'est la formulation adoptée en traitement du signal, par exemple pour la prédiction linéaire par la méthode d'autocorrélation ([38]). Plus généralement, lors de la résolution des systèmes linéaires par les méthodes de Lanczos, les itérations dans les sous-espaces de Krylov s'interprètent comme une évaluation de polynômes en la matrice du système et orthogonaux par rapport à une fonctionnelle qui ne dépend que de la matrice et des conditions initiales de l'itération ([24]). La séquence de Krylov simple dans le cas du gradient conjugué doit être doublée dans la version non symétrique. De nouvelles causes de 'breakdown' apparaissent par rapport au calcul des polynômes orthogonaux ([23]). Remarquons que la progression des méthodes de Lanczos

a suivi celle des méthodes de Hankel que nous décrivons ensuite. Pour formaliser ces connexions nous proposons le schéma synoptique suivant.

Schéma synoptique des connexions

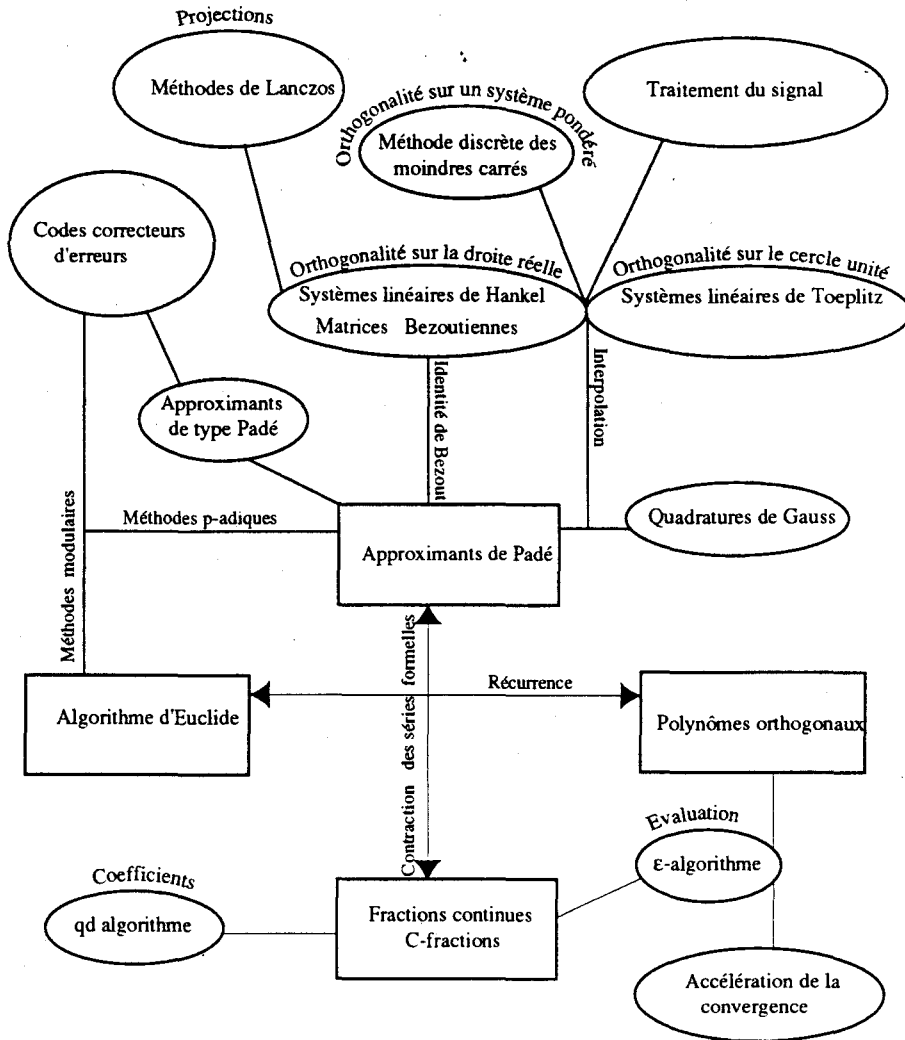


fig. 5

1.5 Problématique

De la présentation générale qui précède nous allons extraire le dénominateur commun. Quelle que soit la tournure du problème (polynomiale, matricielle, rationnelle), le fond des problèmes numériques rencontrés est le même. Stabilité et contrôle du conditionnement, récurrences et sauts, sont les aspects essentiels que partagent les méthodes numériques dans les domaines que nous avons cités. Nous introduisons la problématique au travers de la progression historique des approches effectuées pour résoudre les systèmes de Hankel. L' algorithme présenté par Levinson ([76])(fig.6(1)) a réduit la complexité des systèmes de Toeplitz à $O(n^2)$ en tirant parti des propriétés de déplacement dans la structure de ces matrices. L'extension au cas des matrices de Toeplitz non symétriques et à leur inversion a été proposée par Trench ([102]). Néanmoins, ces algorithmes nécessitent la forte régularité de la matrice. Etendre ces méthodes au cas non fortement régulier a constitué une deuxième étape à laquelle ont contribué Rissanen ([93])(fig.6(2)) et bien d'autres dont on trouvera les références dans ([27]). Le défaut d'irrégularité, alors éliminé en précision exacte, est devenu un problème de quasi-irrégularité en arithmétique de précision finie. Des réponses à ces étapes mal conditionnées ont été récemment présentées. Elles consistent à sauter ces étapes en résolvant des systèmes de type Sylvester (fig.6(3)) par des méthodes classiques du type QR ou pivot de Gauss. L'avantage ainsi acquis concerne surtout la stabilité au détriment de la complexité lorsque les sauts à effectuer sont de grandes tailles. Ces méthodes nécessitent donc de petites bornes pour la taille des sauts afin de conserver une complexité en $O(n^2)$. Obtenir une expression récursive et stable de ces sauts avec une complexité identique est donc un problème important. Les algorithmes de zoom-récursifs présentés dans ([12],[33],[49]) ont la même structure de résolution par blocs. Les points par lesquels ils diffèrent sont la résolution des sauts et les critères de décision dans la taille des sauts à effectuer. L'organigramme commun selon lequel ils sont structurés a la forme suivante.

Organigramme des méthodes itératives
pour la résolution des systèmes de Hankel

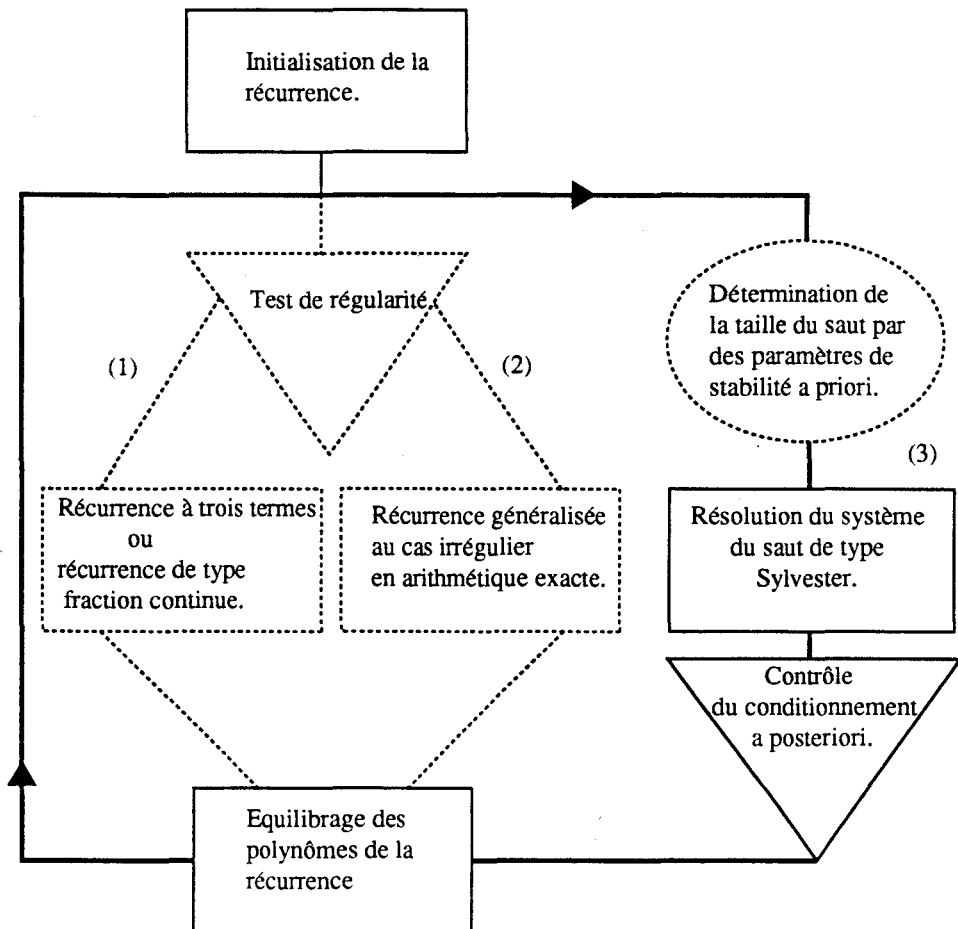


fig. 6

Parallèlement, dans l'ensemble des algorithmes proposés jusqu'à présent, se sont principalement distinguées deux classes de stratégies. On trouve les algorithmes de type Levinson-Trench qui construisent progressivement la factorisation LDU symétrique de H^{-1} , au moyen de la récurrence à trois termes. Ils nécessitent le calcul de produits scalaires et ne sont donc pas adaptés au calcul parallèle. A l'étape n , les produits scalaires nécessaires sont les valeurs résiduelles $c(x^i P_n)$ pour $i = n, n + 1, \dots$ et $c(x^j P_{n-1})$ pour $j = n, n + 1, \dots$. On obtient ces valeurs comme composantes de vecteurs résidus de l'application de H_{2n} sur le vecteur associé à P_n et de même pour P_{n-1} . De la relation de récurrence sur les P_n on déduit une relation de récurrence sur les résidus qui évite le calcul de produits scalaires. Les algorithmes obtenus sont de type Schur, ils utilisent la factorisation LDU symétrique de H liée aux vecteurs résidus. C'est le cas par exemple de l'algorithme proposé dans ([12]). Ils ne nécessitent plus de calcul de produits scalaires et ainsi s'adaptent mieux au calcul parallèle et, dans le cas séquentiel, à l'utilisation de méthodes de Fourier rapides pour le calcul des produits de polynômes. La complexité séquentielle peut alors être réduite à $O(n \log(n)^2)$. Rappelons que dans le problème modèle nous avons présenté les deux méthodes, le développement en fractions continues ne demandant pas de calcul de produit scalaire contrairement à la relation à trois termes. Citons deux autres techniques très récentes, qui n'ont donc pas reçu à ce jour autant d'attention. La première ([64]) repose sur les matrices de Loewner et la seconde se fonde sur l'algorithme de Newton ([84]) pour une adaptation différente au parallélisme. Pour notre part, nous avons développé dans ([91]) une méthode de nature différente puisque reposant sur un calcul régressif dans la table des polynômes orthogonaux. Nous nous proposons d'en donner de multiples interprétations et extensions à travers les chapitres (3,4). En ce qui concerne les algorithmes de zoom récursif, nous avons proposé au chapitre (5) une interprétation des sauts en termes d'approximants de Padé permettant d'envisager ces étapes avec la même complexité $O(n^2)$ que celle des algorithmes de type Levinson sans saut. Les systèmes de Hankel possèdent de nombreuses propriétés dont différents ouvrages exposent des aspects variés. L'étude du rang des matrices de Hankel et des caractéristiques des formes quadratiques ou hermitiennes que l'on peut leur associer est développée dans ([71]). Diverses relations algébriques avec les matrices de Bezout, Vandermonde et matrices de Loewner figurent dans ([67],[46],[64]). Le dernier chapitre nous donne l'occasion de faire le point sur ces connaissances sous une forme inédite et simple grâce à une interprétation polynomiale. Celle-ci se révèle donc un moyen de calcul efficace tant pour étendre par exemple la formule de Gohberg-Semencul ([56]) aux étapes de sauts que pour trouver des prolongements aux méthodes régressives du chapitre (4). Pour introduire les méthodes régressives, nous commençons par étudier quelques aspects des récurrences sur les polynômes orthogonaux et tout particulièrement les récurrences régressives.

Chapitre 2

Diverses présentations algébriques des récurrences

2.1 Introduction

La résolution d'un système linéaire peut être menée par une méthode de bordage ([47]). Dans le cas des systèmes de Hankel, cette méthode se présente sous la forme simplifiée de la récurrence à trois termes. Nous exposons cet aspect dans le paragraphe (2.2) sans l'utilisation des polynômes adjacents, la formulation est donc différente de celle obtenue dans ([21]). Les polynômes orthogonaux adjacents permettent de relier l'algorithme de développement en C -fractions continues et la récurrence à trois termes. Dans le paragraphe (2.3), nous avons également porté notre attention sur l'interprétation géométrique d'une relation de récurrence faisant intervenir les polynômes adjacents. Les relations de récurrences sont utilisées en général sous leur forme progressive, dans le sens de l'augmentation du degré des polynômes. Il est proposé au paragraphe (2.4) une étude algébrique des récurrences régressives. Enfin, nous terminons ce chapitre par la présentation de la récurrence à trois termes sous l'aspect de la matrice de Jacobi.

2.2 La récurrence à trois termes et la méthode de bordage

La récurrence à trois termes (1.8) se démontre simplement en imposant les conditions d'orthogonalité sur une combinaison linéaire des polynômes $xP_k(x)$ et $P_{k-1}(x)$. En dépit de cela, nous allons présenter la récurrence comme résultant d'une méthode de bordage. Les méthodes de bordage consistent à établir les liens existants entre un système linéaire donné (S_k) et un système linéaire de dimension supérieure (S_{k+1}) obtenu en bordant la matrice et le second membre du système de départ (S_k). Cette stratégie a été utilisée avec succès par Levinson ([76]) et Durbin ([58], pp.184-186) pour réduire à $O(n^2)$ la complexité de la résolution des systèmes constitués à partir des matrices de Toeplitz symétriques fortement régulières. Très pratique pour aborder les systèmes ayant des

structures conservées par bordage, la méthode s'est étendue naturellement aux systèmes de Hankel ([102]) puis aux structures par blocs dans ([24]) et ([77]). Il faut également noter que la méthode peut être employée en sens inverse comme dans ([20]). Le système, satisfait par les coefficients du polynôme unitaire $P_k = \sum_{i=0}^{k-1} a_i x^i + x^k$ de degré k et orthogonal par rapport à la fonctionnelle $c = (c_0, c_1, c_2, \dots, c_{2k+1}, \dots)$, peut prendre la forme (1.4),

$$M_k \tilde{p}_k = -u_k,$$

avec $u_k^T = (c_k, c_{k+1}, \dots, c_{2k-1})$, $\tilde{p}_k^T = (a_0, a_1, \dots, a_{k-1})$ et M_k la matrice des moments (1.3) d'ordre k . De plus, on notera $p_k^T = (a_0, a_1, \dots, a_{k-1}, 1)$ pour représenter sous l'aspect vectoriel le polynôme P_k , alors que \tilde{p}_k correspond à P_k tronqué de son terme de plus haut degré. Si on exprime M_{k+1} en bordant M_k , on a également,

$$\begin{aligned} M_{k+1} \begin{pmatrix} \tilde{p}_k \\ 1 \end{pmatrix} &= \begin{pmatrix} M_k & u_k \\ u_k^T & c_{2k} \end{pmatrix} \begin{pmatrix} \tilde{p}_k \\ 1 \end{pmatrix} \\ &= \begin{pmatrix} M_k \tilde{p}_k + u_k \\ u_k^T \tilde{p}_k + c_{2k} \end{pmatrix} \\ &= \begin{pmatrix} o_k \\ u_k^T \tilde{p}_k + c_{2k} \end{pmatrix} \\ &= (u_k^T \tilde{p}_k + c_{2k}) e_{k+1}, \end{aligned}$$

où o_k est le vecteur nul de dimension k et $e_{k+1}^T = (0, 0, \dots, 0, 1)$ de dimension $k+1$. On calcule alors l'inverse de M_{k+1} à l'aide de M_k^{-1} . L'inverse d'une matrice symétrique étant symétrique, il suffit de déterminer la matrice X_k , le vecteur y_k et le nombre α_k tels que

$$\begin{pmatrix} M_k & u_k \\ u_k^T & c_{2k} \end{pmatrix} \begin{pmatrix} X_k & y_k \\ y_k^T & \alpha_k \end{pmatrix} = \begin{pmatrix} I_k & o_k \\ o_k^T & 1 \end{pmatrix}.$$

On obtient le système suivant:

$$\begin{cases} M_k X_k + u_k y_k^T = I_k \\ M_k y_k + \alpha_k u_k = o_k \\ u_k^T X_k + c_{2k} y_k^T = o_k^T \\ u_k^T y_k + \alpha_k c_{2k} = 1. \end{cases}$$

En faisant intervenir M_k^{-1} , on trouve à l'aide de la deuxième équation que l'on reporte dans la première

$$\begin{aligned} y_k &= -\alpha_k M_k^{-1} u_k \\ X_k &= M_k^{-1} + \alpha_k M_k^{-1} u_k u_k^T M_k^{-1}, \end{aligned}$$

puis

$$\alpha_k = \frac{1}{c_{2k} - u_k^T M_k^{-1} u_k}.$$

Ainsi, M_{k+1}^{-1} prend la forme

$$M_{k+1}^{-1} = \begin{pmatrix} M_k^{-1} + \alpha_k M_k^{-1} u_k u_k^T M_k^{-1} & -\alpha_k M_k^{-1} u_k \\ -\alpha_k u_k^T M_k^{-1} & \alpha_k \end{pmatrix}. \quad (2.1)$$

L'expression (2.1) peut se déduire de la formule de Sherman-Morrison-Woodbury ([58], p.51). Cette formule exprime l'inverse de $(M+uv^T)$ pour une matrice M inversible d'ordre n et u et v des vecteurs de dimension n sous la forme

$$(M + uv^T)^{-1} = M^{-1} - M^{-1}u(1 + v^T M^{-1}u)^{-1}v^T M^{-1}.$$

On suppose que $c_{2k} \neq 0$. En appliquant, dans un premier temps, cette formule avec la matrice inversible

$$M = \begin{pmatrix} M_k & o_k \\ o_k^T & c_{2k} \end{pmatrix},$$

et les choix $u = e_{k+1}$ et $v^T = (u_k^T, 0)$, puis une seconde fois avec $u = (u_k^T, 0)^T$ et $v^T = e_{k+1}^T$ on retrouve respectivement l'expression de l'inverse après le bordage inférieur puis après le bordage latéral. Il convient d'observer dans le résultat (2.1) que la restriction $c_{2k} \neq 0$ n'est plus nécessaire. Par contre elle s'est transformée en la condition

$$c_{2k} - u_k^T M_k^{-1} u_k \neq 0.$$

Pour interpréter ceci, rappelons que $M_k^{-1} u_k = -\tilde{p}_k$ et que $c(x^k P_k(x)) = c_{2k} + u_k^T \tilde{p}_k$. Cela permet d'observer que

$$\alpha_k = \frac{1}{c(x^k P_k(x))}$$

et l'on retrouve la condition de régularité de M_{k+1} à partir de la régularité de M_k .

Au rang $k+1$ on a alors

$$\tilde{p}_{k+1} = -M_{k+1}^{-1} u_{k+1}.$$

Soit, en développant à l'aide de (2.1)

$$\tilde{p}_{k+1} = \begin{pmatrix} M_k^{-1} \begin{pmatrix} c_{k+1} \\ \vdots \\ c_{2k} \end{pmatrix} \\ 0 \end{pmatrix} - \frac{1}{c(x^k P_k(x))} \begin{pmatrix} \tilde{p}_k \tilde{p}_k^T & \tilde{p}_k \\ \tilde{p}_k^T & 1 \end{pmatrix} \begin{pmatrix} c_{k+1} \\ \vdots \\ \vdots \\ c_{2k+1} \end{pmatrix}. \quad (2.2)$$

Ceci peut encore se simplifier en notant que

$$c(x^{k+1} P_k) = p_k^T \begin{pmatrix} c_{k+1} \\ \vdots \\ c_{2k+1} \end{pmatrix}$$

et en réorganisant le produit matriciel comme suit

$$\begin{pmatrix} \tilde{p}_k \tilde{p}_k^T & \tilde{p}_k \\ \tilde{p}_k^T & 1 \end{pmatrix} = \begin{pmatrix} \tilde{p}_k \\ 1 \end{pmatrix} \begin{pmatrix} \tilde{p}_k^T & 1 \end{pmatrix}.$$

La relation (2.2) devient alors

$$\begin{aligned}\tilde{p}_{k+1} &= \begin{pmatrix} M_k^{-1} \begin{pmatrix} c_{k+1} \\ \vdots \\ c_{2k} \\ 0 \end{pmatrix} \end{pmatrix} - \frac{1}{c(x^k P_k(x))} \begin{pmatrix} \tilde{p}_k \\ 1 \end{pmatrix} \begin{pmatrix} \tilde{p}_k^T & 1 \end{pmatrix} \begin{pmatrix} c_{k+1} \\ \vdots \\ c_{2k+1} \end{pmatrix} \\ &= \begin{pmatrix} M_k^{-1} \begin{pmatrix} c_{k+1} \\ \vdots \\ c_{2k} \\ 0 \end{pmatrix} \end{pmatrix} - \frac{c(x^{k+1} P_k)}{c(x^k P_k(x))} p_k.\end{aligned}$$

En calculant

$$\begin{aligned}M_k \begin{pmatrix} 0 \\ a_0 \\ \vdots \\ a_{k-2} \end{pmatrix} &= \begin{pmatrix} c_1 a_0 + \cdots + c_{k-1} a_{k-2} \\ c_2 a_0 + \cdots + c_k a_{k-2} \\ \vdots \\ c_k a_0 + \cdots + c_{2k-2} a_{k-2} \end{pmatrix} \\ &= \begin{pmatrix} c_1 a_0 + \cdots + c_{k-1} a_{k-2} + c_k a_{k-1} \\ c_2 a_0 + \cdots + c_k a_{k-2} + c_{k+1} a_{k-1} \\ \vdots \\ c_k a_0 + \cdots + c_{2k-2} a_{k-2} + c_{2k-1} a_{k-1} \end{pmatrix} - a_{k-1} \begin{pmatrix} c_k \\ \vdots \\ c_{2k-1} \end{pmatrix} \\ &= \begin{pmatrix} -c_{k+1} \\ \vdots \\ -c_{2k-1} \\ c(x^k P_k(x)) - c_{2k} \end{pmatrix} - a_{k-1} \begin{pmatrix} c_k \\ \vdots \\ c_{2k-1} \end{pmatrix} \\ &= - \begin{pmatrix} c_{k+1} \\ \vdots \\ c_{2k-1} \\ c_{2k} \end{pmatrix} - a_{k-1} \begin{pmatrix} c_k \\ \vdots \\ c_{2k-1} \end{pmatrix} + c(x^k P_k(x)) e_k,\end{aligned} \tag{2.3}$$

puis en appliquant M_k^{-1} à (2.3) on peut extraire les égalités suivantes

$$\begin{aligned}-M_k^{-1} \begin{pmatrix} c_{k+1} \\ \vdots \\ c_{2k-1} \\ c_{2k} \end{pmatrix} &= \begin{pmatrix} 0 \\ a_0 \\ \vdots \\ a_{k-2} \end{pmatrix} + a_{k-1} M_k^{-1} \begin{pmatrix} c_k \\ \vdots \\ c_{2k-1} \end{pmatrix} - c(x^k P_k(x)) M_k^{-1} e_k \\ &= \begin{pmatrix} 0 \\ a_0 \\ \vdots \\ a_{k-2} \end{pmatrix} - a_{k-1} \tilde{p}_k - \frac{c(x^k P_k(x))}{c(x^{k-1} P_{k-1}(x))} p_{k-1}.\end{aligned}$$

En reportant finalement dans (2.2) cela donne

$$\tilde{p}_{k+1} = \begin{pmatrix} 0 \\ a_0 \\ \vdots \\ a_{k-2} \\ 0 \end{pmatrix} - a_{k-1} \begin{pmatrix} \tilde{p}_k \\ 0 \end{pmatrix} - \frac{c(x^k P_k(x))}{c(x^{k-1} P_{k-1}(x))} \begin{pmatrix} p_{k-1} \\ 0 \end{pmatrix} - \frac{c(x^{k+1} P_k(x))}{c(x^k P_k(x))} p_k.$$

En combinant les dernières composantes des deux premiers vecteurs, on obtient

$$\tilde{p}_{k+1} = \begin{pmatrix} 0 \\ a_0 \\ \vdots \\ a_{k-2} \\ a_{k-1} \end{pmatrix} - a_{k-1} p_k - \frac{c(x^k P_k(x))}{c(x^{k-1} P_{k-1}(x))} \begin{pmatrix} p_{k-1} \\ 0 \end{pmatrix} - \frac{c(x^{k+1} P_k(x))}{c(x^k P_k(x))} p_k.$$

On complète et regroupe les termes de la relation précédente pour faire apparaître p_{k+1} et la relation à trois termes

$$p_{k+1} = \begin{pmatrix} \tilde{p}_{k+1} \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ a_0 \\ \vdots \\ a_{k-1} \\ 1 \end{pmatrix} - \left(a_{k-1} + \frac{c(x^{k+1} P_k(x))}{c(x^k P_k(x))} \right) \begin{pmatrix} p_k \\ 0 \end{pmatrix} - \frac{c(x^k P_k(x))}{c(x^{k-1} P_{k-1}(x))} \begin{pmatrix} p_{k-1} \\ 0 \\ 0 \end{pmatrix},$$

c'est-à-dire, en adoptant l'écriture polynomiale

$$P_{k+1}(x) = \left(x - \left(a_{k-1} + \frac{c(x^{k+1} P_k(x))}{c(x^k P_k(x))} \right) \right) P_k(x) - \frac{c(x^k P_k(x))}{c(x^{k-1} P_{k-1}(x))} P_{k-1}(x).$$

Pour identifier (1.8), on calcule a_{k-1} en exprimant l'orthogonalité de P_{k-1} et P_k

$$0 = c(P_{k-1} P_k) = c(x^k P_{k-1}) + a_{k-1} c(x^{k-1} P_{k-1}) + \dots \text{des termes nuls par orthogonalité.}$$

D'où il ressort que

$$a_{k-1} = -\frac{c(x^k P_{k-1})}{c(x^{k-1} P_{k-1})}.$$

Bien d'autres relations entre des familles de polynômes orthogonaux peuvent apparaître par des méthodes de bordage. C'est le cas par exemple des polynômes adjacents ([39], p. 97) que nous étudions à présent.

2.3 Les polynômes adjacents et autres relations de récurrence

Les polynômes P_k de (1.8) orthogonaux par rapport à la fonctionnelle $c = (c_0, c_1, \dots)$ peuvent être exprimés comme rapport de deux déterminants ([60],[15])

$$P_k(x) = \frac{\begin{vmatrix} c_0 & c_1 & \cdots & c_{k-1} & c_k \\ c_1 & c_2 & \cdots & c_k & c_{k+1} \\ \vdots & \vdots & & \vdots & \vdots \\ c_{k-1} & c_k & \cdots & c_{2k-2} & c_{2k-1} \\ 1 & x & \cdots & x^{k-1} & x^k \end{vmatrix}}{\begin{vmatrix} c_0 & c_1 & \cdots & c_{k-2} & c_{k-1} \\ c_1 & c_2 & \cdots & c_{k-1} & c_k \\ \vdots & \vdots & & \vdots & \vdots \\ c_{k-2} & c_{k-1} & \cdots & c_{2k-4} & c_{2k-3} \\ c_{k-1} & c_k & \cdots & c_{2k-3} & c_{2k-2} \end{vmatrix}}. \quad (2.4)$$

Les dénominateurs sont les mineurs principaux de la matrice des moments. Si on extrait une matrice de M_k en supprimant les i dernières lignes et les i premières colonnes de M_k on obtient encore une matrice de Hankel. Elle définit une nouvelle matrice de moments et engendre sa propre famille de polynômes orthogonaux. Les différentes familles produites de cette façon s'obtiennent également par décalages de la fonctionnelle de départ. Nous exposons maintenant les rapports établis dans ([15]) et généralisés dans ([39]), entre les familles de polynômes orthogonaux caractérisées par des diagonales adjacentes.

Soit M_{k+2} la matrice de Hankel d'ordre $k+2$ associée à la fonctionnelle c ,

$$M_{k+2} = \begin{pmatrix} c_0 & c_1 & \cdots & c_k & c_{k+1} \\ c_1 & c_2 & \cdots & c_{k+1} & c_{k+2} \\ \vdots & \vdots & & \vdots & \vdots \\ c_k & c_{k+1} & \cdots & c_{2k} & c_{2k+1} \\ c_{k+1} & c_{k+2} & \cdots & c_{2k+1} & c_{2k+2} \end{pmatrix}.$$

Considérons la matrice notée $M_{k+1}^{(1)}$, obtenue à partir de la matrice M_{k+2} en supprimant la première colonne et la dernière ligne. En tant que matrice de moments elle permet de définir, par la relation de récurrence à trois termes (1.8) appropriée, la famille des polynômes $P_i^{(1)}$ orthogonaux par rapport à la fonctionnelle $c^{(1)} = (c_1, c_2, \dots)$. Pour relier les familles adjacentes P_i et $P_i^{(1)}$, nous décrirons leurs calculs à partir de systèmes linéaires aussi proches que possible, au sens des méthodes de bordage.

Pour k fixé, si le polynôme $P_k^{(1)}(x) = \sum_{i=0}^k b_i x^i$ existe, nous choisissons la normalisation

unitaire ($b_k = 1$). On a donc le système suivant

$$\begin{pmatrix} c_1 & c_2 & \cdots & c_{k-1} & c_k \\ c_2 & c_3 & \cdots & c_k & c_{k+1} \\ \vdots & \vdots & & \vdots & \vdots \\ c_{k-1} & c_k & \cdots & c_{2k-3} & c_{2k-2} \\ c_k & c_{k+1} & \cdots & c_{2k-2} & c_{2k-1} \end{pmatrix} \begin{pmatrix} b_0 \\ b_1 \\ \vdots \\ \vdots \\ b_{k-1} \end{pmatrix} = - \begin{pmatrix} c_{k+1} \\ c_{k+2} \\ \vdots \\ \vdots \\ c_{2k} \end{pmatrix}. \quad (2.5)$$

On peut également l'écrire

$$\begin{pmatrix} c_{2k-1} & c_{2k-2} & \cdots & c_{k+1} & c_k \\ c_{2k-2} & c_{2k-3} & \cdots & c_k & c_{k-1} \\ \vdots & \vdots & & \vdots & \vdots \\ c_{k+1} & c_k & \cdots & c_3 & c_2 \\ c_k & c_{k-1} & \cdots & c_2 & c_1 \end{pmatrix} \begin{pmatrix} b_{k-1} \\ b_{k-2} \\ \vdots \\ \vdots \\ b_0 \end{pmatrix} = - \begin{pmatrix} c_{2k} \\ c_{2k-1} \\ \vdots \\ \vdots \\ c_{k+1} \end{pmatrix}. \quad (2.6)$$

Sous cette forme, nous observons que le second membre complète la matrice du système par la gauche. Parallèlement, pour que le système auquel obéissent les composantes du polynôme $\widetilde{P}_k = \sum_{i=0}^k a_i x^i$, de degré k orthogonal par rapport à la fonctionnelle c soit basé sur la même matrice que celle du système (2.5), il faut imposer la contrainte de normalisation $\widetilde{P}_k(0) = 1$. En effet, le système imposé avant tout choix de normalisation étant

$$c(x^i \widetilde{P}_k) = 0, \quad \forall i \leq k-1$$

\Leftrightarrow

$$\begin{pmatrix} c_0 & c_1 & c_2 & \cdots & c_{k-1} & c_k \\ c_1 & c_2 & c_3 & \cdots & c_k & c_{k+1} \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ c_{k-2} & c_{k-1} & c_k & \cdots & c_{2k-3} & c_{2k-2} \\ c_{k-1} & c_k & c_{k+1} & \cdots & c_{2k-2} & c_{2k-1} \\ c_k & c_{k+1} & \cdots & \cdots & c_{2k-1} & c_{2k} \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ \vdots \\ a_{k-1} \\ a_k \end{pmatrix} = \begin{pmatrix} c(\widetilde{P}_k) \\ c(x\widetilde{P}_k) \\ \vdots \\ \vdots \\ c(x^{k-1}\widetilde{P}_k) \\ c(x^k\widetilde{P}_k) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ \vdots \\ 0 \\ c(x^k\widetilde{P}_k) \end{pmatrix},$$

en choisissant la contrainte de normalisation $a_0 = 1$ pour libérer la première colonne de la matrice, on obtient

$$\begin{pmatrix} c_1 & c_2 & \cdots & c_{k-1} & c_k \\ c_2 & c_3 & \cdots & c_k & c_{k+1} \\ \vdots & \vdots & & \vdots & \vdots \\ c_{k-1} & c_k & \cdots & c_{2k-3} & c_{2k-2} \\ c_k & c_{k+1} & \cdots & c_{2k-2} & c_{2k-1} \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ \vdots \\ a_k \end{pmatrix} = - \begin{pmatrix} c_0 \\ c_1 \\ \vdots \\ \vdots \\ c_{k-1} \end{pmatrix}. \quad (2.7)$$

Désormais, on désignera par normalisation en zéro la normalisation $P_k(0) = 1$. On peut alors constater que le second membre complète la matrice du système par la gauche et ainsi établir un lien de symétrie avec les polynômes adjacents (2.6).

Propriété 1 *Pour la fonctionnelle tonquée $c = (c_0, c_1, \dots, c_{2k})$, le polynôme adjacent unitaire $P_k^{(1)}$ de degré k devient, une fois l'ordre de ses coefficients renversé, le polynôme orthogonal relativement à la fonctionnelle renversée $\tilde{c} = (c_{2k}, c_{2k-1}, \dots, c_0)$, normalisé en zéro.*

Un polynôme adjacent se révèle être, en le renversant, un polynôme orthogonal pour la fonctionnelle renversée. Inversement, en renversant un polynôme orthogonal normalisé en zéro, on obtient un polynôme adjacent pour la fonctionnelle renversée. En prenant un peu de recul, les polynômes orthogonaux décrivent le développement de la fonctionnelle de la gauche vers la droite. La propriété précédente souligne que les polynômes adjacents décrivent le développement de la fonctionnelle de la droite vers la gauche.

Contrairement aux polynômes P_k les polynômes \widetilde{P}_k traduisent le prolongement à gauche de la fonctionnelle c et de la matrice de Hankel, le second membre formant la colonne précédente de la matrice de Hankel au lieu de former la colonne suivante. Les systèmes (2.7) et (2.5) basés sur la même matrice de Hankel seront donc réguliers simultanément, assurant ainsi l'existence simultanée de \widetilde{P}_k et $P_k^{(1)}$. L'intérêt majeur des polynômes adjacents $P_k^{(1)}$ est de fournir des relations de récurrences reliant les deux familles, et de permettre ainsi d'autres méthodes numériques de calculs pour les polynômes orthogonaux, certaines formes pouvant se montrer plus stables que d'autres. En arithmétique exacte, les relations de récurrences ne sont pas non plus équivalentes. Il se peut en effet que l'un des $P_k^{(1)}$ n'existe pas sans pour autant que les polynômes orthogonaux soient irréguliers. Ce n'est le cas que lorsque $P_k(0) = 0$, car alors \widetilde{P}_k , la normalisation en zéro de P_k , n'existe pas. Cela entraîne l'inexistence de $P_k^{(1)}$ comme on vient de le voir. Regardons l'exemple suivant.

Exemple 7

Pour la fonctionnelle $c := (1, 1, 2, 3, 4, 5, 5, 4, 3, 2, \dots)$, les premiers polynômes orthogonaux sont réguliers.

$$\begin{aligned} P_0 &= 1, \\ P_1 &= t - 1, \\ P_2 &= t^2 - t - 1, \\ P_3 &= t^3 - 2t^2 + t, \\ P_4 &= t^4 - t^3 - 2t^2 + 2t + 1, \\ P_5 &= t^5 - 2t^4 + 2t^3 - 2t^2 + 2t - 1. \end{aligned}$$

Pourtant, en utilisant la relation de récurrence sur les polynômes adjacents

$$P_0^{(1)} = 1,$$

$$\begin{aligned} P_1^{(1)} &= t - 2, \\ P_2^{(1)} &= t^2 - 2t + 1, \end{aligned}$$

et $P_3^{(1)}$ est irrégulier. La récurrence mixte, utilisant les polynômes orthogonaux et leurs adjacents est dans ce cas confrontée à une irrégularité.

◇

Pour les mêmes raisons, un mauvais conditionnement peut très bien intervenir sur une récurrence avec les $P_k^{(1)}$ sans paraître dans la récurrence à trois termes satisfaite par les P_k .

Les familles de polynômes P_k et $P_k^{(1)}$ vérifient des relations croisées que l'on peut obtenir par divers procédés. Nous en présentons ici une approche géométrique, celle-ci n'apparaissant pas toujours clairement lorsque l'on se restreint à l'un des points de vue polynomial ou matriciel.

$P_k^{(1)}$ est soumis par la fonctionnelle $c^{(1)}$ aux contraintes

$$c^{(1)}(x^i P_k^{(1)}) = 0 \text{ pour } i = 0, \dots, k - 1$$

que l'on traduit sur la fonctionnelle c puis le polynômes $xP_k^{(1)}$ de degré $k + 1$ par

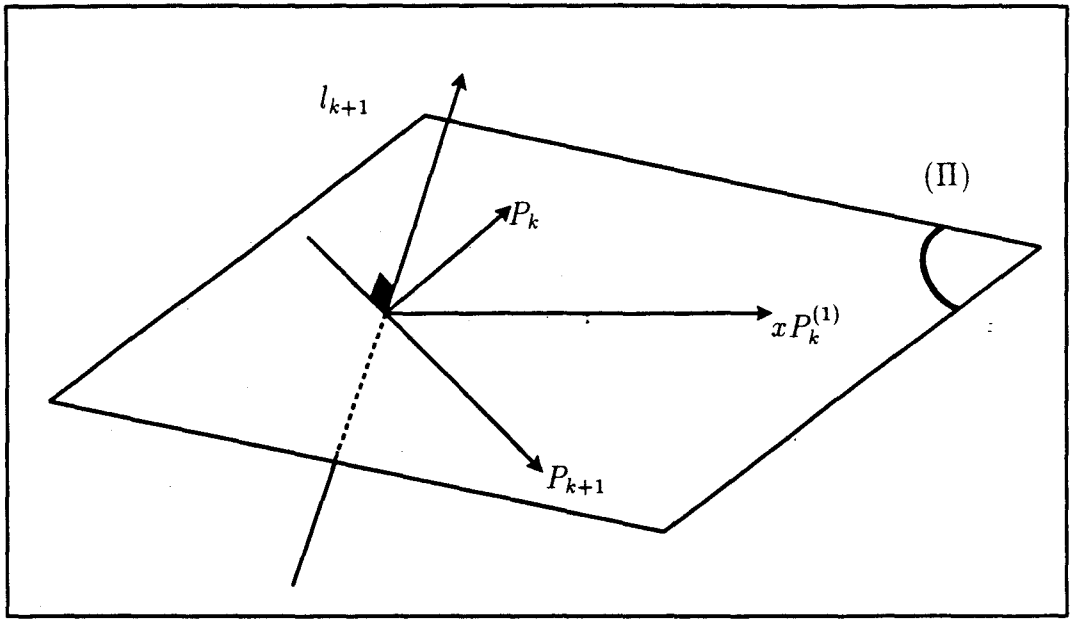
$$c(x^{i+1} P_k^{(1)}) = 0 \text{ pour } i = 0, \dots, k - 1$$

⇔

$$c(x^i (xP_k^{(1)})) = 0 \text{ pour } i = 0, \dots, k - 1.$$

Ces conditions d'orthogonalité par rapport à la fonctionnelle c , jusqu'au rang $k - 1$, sont également vérifiées par le polynôme P_k . Ces conditions s'interprètent vectoriellement comme l'orthogonalité du vecteur dont les composantes sont les coefficients de $xP_k^{(1)}$ aux k premiers vecteurs lignes de la matrice de Hankel M_{k+2} d'ordre $k + 2$. Si l'on considère le polynôme P_k (prolongé par une composante nulle correspondant au coefficient sur le degré $k + 1$) et $xP_k^{(1)}$ comme des vecteurs de dimension $k + 2$, le plan vectoriel qu'ils engendrent est constitué de vecteurs satisfaisant les contraintes linéaires précédentes. L'espace vectoriel V_{k+2} auquel ces vecteurs appartiennent est de dimension $k + 2$ et il contient le vecteur associé à P_{k+1} ainsi que le vecteur, noté l_{k+1} , constitué par la $k + 1$ -ième ligne de la matrice M_{k+2} . Nous représentons ces éléments sur la figure suivante.

fig. 7



Le polynôme P_{k+1} cherché doit être de degré $k + 1$ et satisfaire les deux conditions suivantes:

$$\begin{cases} P_{k+1} \text{ vérifie les mêmes conditions d'orthogonalité que les vecteurs de } (\Pi) \\ P_{k+1} \text{ est orthogonal à } l_{k+1}. \end{cases} \quad (2.8)$$

Dans l'espace vectoriel V_{k+2} , l'hyperplan (T) orthogonal à l_{k+1} coupe le plan (Π) suivant un espace vectoriel de dimension au moins un. Pour une raison de dimensions, deux cas se présentent:

1. $(\Pi) \cap (T)$ est le plan (Π) . Cela signifie en particulier que le vecteur associé à P_k est orthogonal à l_{k+1} , c'est-à-dire que $c(x^k P_k) = 0$ et P_{k+1} n'existe pas. On retrouve le cas irrégulier.
2. $(\Pi) \cap (T)$ est une droite vectorielle (L) . Alors, n'importe quel vecteur non nul de (L) vérifie les deux conditions de (2.8). Si $P_k \in (L)$ alors on retombe dans le cas d'irrégularité précédent. Enfin, si $P_k \notin (L)$ alors on dispose d'une combinaison linéaire non nulle de P_k et $xP_k^{(1)}$ respectant toutes les conditions requises pour définir P_{k+1} , celle du degré comprise (voir la figure (7)). Pour respecter le caractère unitaire de P_{k+1} la combinaison doit être de la forme

$$P_{k+1} = xP_k^{(1)} + \lambda P_k. \quad (2.9)$$

Le calcul du coefficient λ provient de l'orthogonalité de P_{k+1} et l_{k+1} . On ne peut obtenir λ qu'à la seule condition $c(x^k P_k) \neq 0$, alors

$$\lambda = -\frac{c(x^{(k+1)} P_k^{(1)})}{c(x^k P_k)}.$$

La visualisation géométrique nous donne une idée des conditions de stabilité de cette relation de récurrence. La stabilité de la définition de (Π) est directement conditionnée par l'angle formé par les vecteurs normalisés associés à P_k et $xP_k^{(1)}$. Une ouverture angulaire proche de l'angle droit signifiera une meilleure précision des combinaisons linéaires. On peut donner une mesure numérique de cet écart angulaire par son cosinus:

$$\cos(\theta) = \frac{\langle P_k, xP_k^{(1)} \rangle}{\|P_k\| \|xP_k^{(1)}\|} \quad (2.10)$$

où $\langle \cdot, \cdot \rangle$ et $\|\cdot\|$ désignent respectivement le produit scalaire Euclidien et la norme Euclidienne des vecteurs de \mathbb{R}^{k+2} . Cela se généralise au cas complexe en prenant la norme et le produit Hermitien de \mathbb{C}^{k+2} . Remarquons que P_k et $xP_k^{(1)}$ ont des vecteurs associés non colinéaires, les degrés des polynômes étant différents. Si θ n'est pas nul, il peut cependant être très petit. D'autre part, la projection du vecteur l_{k+1} est d'autant mieux définie que ce vecteur s'éloigne de la perpendiculaire à (Π) d'après le cas 1. On peut considérer la condition précédente comme un cas particulier d'orthogonalité de l_{k+1} et P_k , qui d'après le cas 2 conduit à l'irrégularité. En désignant par ϕ une mesure de l'angle (P_k, l_{k+1}) , alors plus la quantité

$$\cos(\phi) = \frac{\langle P_k, l_{k+1} \rangle}{\|P_k\| \|l_{k+1}\|} \quad (2.11)$$

est proche de zéro et plus on aura d'imprécision sur P_{k+1} . Rappelons que $P_{k+1}^{(1)}$ et \widetilde{P}_{k+1} sont réguliers simultanément, et sauf si $P_{k+1}(0) = 0$, l'irrégularité de $P_{k+1}^{(1)}$ impose celle de P_{k+1} . Or ceci advient dès que l_{k+1} et $xP_k^{(1)}$ sont orthogonaux. Par suite, la source d'imprécision relevant des positions relatives de l_{k+1} et $xP_k^{(1)}$ est observée au travers d'une mesure de l'angle entre $xP_k^{(1)}$ et l_{k+1} .

$$\cos(\psi) = \frac{\langle xP_k^{(1)}, l_{k+1} \rangle}{\|xP_k^{(1)}\| \|l_{k+1}\|} \quad (2.12)$$

Le cas $P_{k+1}(0) = 0$ relève de l'irrégularité de la fonctionnelle renversée. En effet si $P_{k+1}(0) = 0$, de l'unicité de P_{k+1} à un coefficient de normalisation près, on déduit l'impossibilité d'obtenir $P_{k+1}(0) = 1$ par normalisation. Ainsi \widetilde{P}_{k+1} n'existe pas, entraînant simultanément l'inexistence de $P_{k+1}^{(1)}$. Or, en appliquant la propriété (1), l'inexistence de $P_{k+1}^{(1)}$ se traduit par l'irrégularité de la fonctionnelle renversée. Tout ceci permet d'illustrer à nouveau la parfaite symétrie dans laquelle P_{k+1} et $P_{k+1}^{(1)}$ coexistent. Dans le cas où $P_{k+1}^{(1)}$ n'existe pas, la fonctionnelle renversée est irrégulière, on peut donc tout naturellement qualifier la fonctionnelle d'irrégulière à gauche. Nous utiliserons la symétrie de la propriété 1 lorsque nous étudierons le cas des fonctionnelles symétriques (i.e $c_i = c_{n-i}$ pour $i = 0, \dots, n$). Dans le cas général, le calcul récursif des polynômes $P_k^{(1)}$ et P_k nécessite une autre relation de récurrence, par exemple

$$P_{k+1}^{(1)} = (x + \alpha)P_k^{(1)} - \beta P_k. \quad (2.13)$$

En imposant les conditions d'orthogonalité $c(x^{k+1}P_{k+1}^{(1)}) = 0$ et $c(x^kP_{k+1}^{(1)}) = 0$ on obtient les coefficients

$$\alpha = \frac{c(x^{k+1}P_k^{(1)})c(x^{k+1}P_k) - c(x^{k+2}P_k^{(1)})c(x^kP_k)}{c(x^{k+1}P_k^{(1)})c(x^kP_k)}$$

et

$$\beta = c(x^{k+1}P_k^{(1)})/c(x^kP_k).$$

En mettant en œuvre les deux relations de récurrence simultanément on obtient l'ossature d'une méthode pour le calcul des polynômes orthogonaux. Notons au passage que dans le cadre plus général des polynômes biorthogonaux, ces relations se sont montrées équivalentes à la méthode de bordage ([21]); elles représentent une forme décomposée de la récurrence à trois termes. Nous sommes à présent en mesure de compléter la liaison entre les deux algorithmes de calculs d'approximants de Padé d'une série formelle $f(x)$ que nous avons exposés au chapitre (1). Rappelons que le premier calculait les dénominateurs des approximants de Padé de type $[n/(n+1)]_f$ comme des polynômes orthogonaux par rapport à la fonctionnelle c définie par les coefficients de la série formelle f . Le deuxième algorithme employait la décomposition en fractions continues de f et faisait intervenir alternativement les deux formes d'approximants de Padé $[n/(n+1)]_f$ et $[n/n]_f$ pour la même fonctionnelle c . Or les polynômes adjacents par rapport à c sont reliés aux dénominateurs de l'approximant $[n/n]$ par la formule suivante ([15])

$$[n/n]_f = x \frac{\widetilde{Q}_n^{(1)}}{P_n^{(1)}},$$

où $\widetilde{P}_n^{(1)}$ est le polynôme renversé du polynôme adjacent $P_n^{(1)}$ et $\widetilde{Q}_n^{(1)}$ désigne le renversé de l'associé de $P_n^{(1)}$ pour la fonctionnelle $c^{(1)}$. Les relations (2.9) et (2.13) sont les renversées des récurrences que l'on met en œuvre dans l'algorithme de calcul récurrent des dénominateurs des développements en C -fractions. Le parcours en escalier décrit par l'algorithme dans la table de Padé se retrouve donc dans la table des polynômes orthogonaux. Ces récurrences sont par exemple la base de l'algorithme *NPADÉ* ([33]). Nous soulignons que la différence entre le calcul des approximants de Padé et celui des polynômes orthogonaux ne se manifeste qu'en présence des cas $\widetilde{P}_k(0) = 0$ ou $P_k(0) = 0$ et que ces inconvénients symétriques l'un de l'autre proviennent respectivement d'une étape de bordage de la matrice par la gauche et par la droite. Nous serons amenés à étudier l'influence des mesures de chaque angle sur la stabilité numérique de ces récurrences. Bien que d'autres relations de récurrences puissent donner d'autres méthodes de progressions pour le calcul des polynômes orthogonaux (voir ([15])), nous nous restreignons à ces deux exemples qui serviront également de modèle lors de la présentation de méthodes régressives. Afin de visualiser l'ensemble des relations disponibles, nous allons décrire la table des polynômes orthogonaux.

2.4 La table des polynômes orthogonaux

Les polynômes adjacents ont eux-mêmes des adjacents de sorte qu'à un décalage de la fonctionnelle de i rangs correspond une famille orthogonale notée $\{P_k^{(i)}, k \in \mathbb{N}\}$. L'ensemble des polynômes orthogonaux et de leurs adjacents se représentent dans une table triangulaire de la forme suivante,

$$\begin{array}{ccccccc}
 P_0 & P_0^{(1)} & P_0^{(2)} & \dots & P_0^{(i)} & P_0^{(i+1)} & \dots \\
 & P_1 & P_1^{(1)} & \dots & P_1^{(i-1)} & P_1^{(i)} & \dots \\
 & & P_2 & \dots & P_2^{(i-2)} & P_2^{(i-1)} & \dots \\
 & & & \ddots & \ddots & \ddots & \ddots \\
 & & & & P_i & P_i^{(1)} & \dots \\
 & & & & & \ddots & \ddots
 \end{array} \tag{2.14}$$

La présentation usuellement adoptée est triangulaire inférieure de manière à assurer la cohérence avec la présentation de la table des approximants de Padé. Cela ne modifie en rien les remarques suivantes.

Remarque 1 Soit H_∞ la matrice de Hankel de dimension infinie engendrée par c . Si l'on superpose la table à la partie triangulaire supérieure de H_∞ alors la position de chaque polynôme orthogonal de la table coïncide avec la position du coin inférieur droit de la sous matrice de H_∞ de dimension finie qui le détermine (voir la figure (6.9.6)). Cela situe donc immédiatement les polynômes concernés par chaque sous matrice de H_∞ .

Remarque 2 Cela permet ensuite de traduire les relations de récurrences en chemins parcourus dans la table ou dans l'ensemble des sous matrices de H_∞ .

Un point important, et qui concernera directement les méthodes régressives, est de décrire les relations de récurrences qui permettent de remonter dans la table et de distinguer celles qui nécessitent la connaissance de moments. Répérorions les relations nécessaires à l'obtention de la table à partir de deux éléments consécutifs d'une diagonale de la table. Les déplacements le long des diagonales descendantes sont décrits par la récurrence à trois termes. On symbolise la relation sous la forme

$$P_{n-1}^{(i)}, P_n^{(i)} \longrightarrow P_{n+1}^{(i)}.$$

Pour remonter la diagonale, on doit utiliser la relation dans l'autre sens.

$$P_{n+1}^{(i)}, P_n^{(i)} \longrightarrow P_{n-1}^{(i)}.$$

Nous décrirons cette relation plus précisément au paragraphe (3.2). Disons seulement que l'interprétation de cette dernière relation en termes de division euclidienne dispense de

la connaissance des moments lors du calcul de $P_{n-1}^{(i)}$ à partir des deux autres polynômes. Pour se déplacer verticalement dans la table, on doit réunir plusieurs relations.

$$\begin{cases} P_{n+1}^{(i)}, P_n^{(i+1)} \longrightarrow P_n^{(i)} \\ P_n^{(i)}, P_n^{(i+1)} \longrightarrow P_{n-1}^{(i+1)} \\ P_n^{(i+1)}, P_{n-1}^{(i+1)} \longrightarrow P_{n-1}^{(i+2)} \end{cases} \Rightarrow P_{n+1}^{(i)}, P_n^{(i+1)} \longrightarrow P_{n-1}^{(i+2)}.$$

La première relation provient de (2.9) prise en sens inverse, soit

$$P_{n+1}^{(i)} - xP_n^{(i+1)} = \lambda P_n^{(i)}, \quad (2.15)$$

avec λ qui se déduit du fait que $P_n^{(i)}$ est unitaire. Cette relation n'utilise pas les moments de la fonctionnelle. Pour la deuxième relation, on constate simplement que $P_n^{(i)}$ vérifie les conditions d'orthogonalité

$$c^{(i)}(x^s P_n^{(i)}) = 0 \text{ pour } s = 0, \dots, n-1,$$

ou encore,

$$c^{(i+1)}(x^s P_n^{(i)}) = 0 \text{ pour } s = 0, \dots, n-2.$$

Comme $P_n^{(i+1)}$ vérifie

$$c^{(i+1)}(x^s P_n^{(i+1)}) = 0 \text{ pour } s = 0, \dots, n-1,$$

la combinaison linéaire de $P_n^{(i+1)}$ et $P_n^{(i)}$ qui donne un polynôme de degré $n-1$ vérifie les conditions d'orthogonalité jusqu'à $n-2$, cela donne donc, à la normalisation près, $P_{n-1}^{(i+1)}$. On calcule donc

$$\eta P_{n-1}^{(i+1)} = P_n^{(i)} - P_n^{(i+1)}$$

où η est imposé par la normalisation unitaire de $P_{n-1}^{(i+1)}$. A nouveau la relation est calculée indépendamment des moments de la fonctionnelle. Enfin, il en est de même pour la troisième relation. Egalement basée sur (2.9), elle s'écrit

$$xP_{n-1}^{(i+2)} = P_n^{(i+1)} - \delta P_{n-1}^{(i+1)},$$

où δ doit rendre la combinaison nulle en $x=0$. En associant les trois relations, il est donc possible de construire la relation $P_{n+1}^{(i)}, P_n^{(i+1)} \longrightarrow P_{n-1}^{(i+2)}$ et de remonter une verticale de la table sans utiliser les moments. On obtient en effet,

$$xP_{n-1}^{(i+2)} = (x + \alpha)P_n^{(i+1)} - P_{n+1}^{(i)}$$

avec pour choix de α celui qui rend la combinaison nulle pour $x=0$.

On se donne maintenant deux polynômes orthogonaux consécutifs $P_j^{(i)}, P_{j+1}^{(i)}$ sur une diagonale notée $d^{(i)}$. D'après ce qui précède, il est possible de déterminer, sans l'aide des moments, la partie triangulaire de la table constituée des polynômes de la diagonale $d^{(i)}$ qui sont de degré inférieur à ceux de départ et de tous ceux qui se situent au-dessus. Cette partie triangulaire admet pour sommets les polynômes $P_{j+1}^{(i)}, P_0^{(i)}$ et $P_0^{(i+j+1)}$.

$$\begin{array}{cccccccc}
P_0 & P_0^{(1)} & \dots & P_0^{(i)} & \dots & \dots & P_0^{(i+j+1)} & \dots \\
& P_1 & \ddots & & \ddots & & \vdots & \\
& & \ddots & \ddots & \vdots & P_j^{(i)} & P_j^{(i+1)} & \\
& & & \ddots & \ddots & & P_{j+1}^{(i)} & \\
& & & & \ddots & \ddots & & \ddots
\end{array} \tag{2.16}$$

La question est de savoir alors si l'on peut obtenir davantage sans les moments et sinon de préciser la partie de la fonctionnelle dont on a besoin pour obtenir plus de polynômes. Sans la fonctionnelle, il est naturellement impossible d'obtenir des polynômes qui sont déterminés par des moments ne faisant pas intégralement partie de ceux qui sont implicitement contenus dans la donnée de $P_j^{(i)}, P_{j+1}^{(i)}$. En utilisant la remarque (1), on constate donc l'impossibilité de calculer sans connaître de moments supplémentaires (comme c_{i-1}) les polynômes $P_s^{(k)}$ tels que $k < i$. Décrivons les matrices qui caractérisent trois polynômes successifs le long d'une diagonale montante, par exemple $P_j^{(i)}, P_{j-1}^{(i+2)}$ et $P_{j-2}^{(i+4)}$.

$$H_\infty = \left(\begin{array}{cccccccccccc}
\dots & c_i & c_{i+1} & c_{i+2} & c_{i+3} & c_{i+4} & \dots & c_{i+j} & c_{i+j+1} & c_{i+j+2} & \dots \\
\dots & c_{i+1} & c_{i+2} & c_{i+3} & c_{i+4} & c_{i+5} & \dots & c_{i+j+1} & c_{i+j+2} & c_{i+j+3} & \dots \\
\dots & \vdots & \vdots & \vdots & \vdots & \vdots & \dots & \vdots & \vdots & \vdots & \dots \\
\dots & \vdots & \vdots & \vdots & \vdots & \vdots & \dots & \vdots & \vdots & \vdots & \dots \\
\dots & \vdots & \vdots & \vdots & \vdots & \vdots & \dots & \vdots & \vdots & \vdots & \dots \\
\dots & c_{i+j-2} & c_{i+j-1} & c_{i+j} & c_{i+j+1} & c_{i+j+2} & \dots & c_{i+2j-2} & c_{i+2j-1} & c_{i+2j} & \dots \\
\dots & c_{i+j-1} & c_{i+j} & c_{i+j+1} & c_{i+j+2} & c_{i+j+3} & \dots & c_{i+2j-1} & c_{i+2j} & \bullet & P_{j-2}^{(i+4)} \\
\dots & c_{i+j} & c_{i+j+1} & c_{i+j+2} & c_{i+j+3} & c_{i+j+4} & \dots & c_{i+2j} & \bullet & P_{j-1}^{(i+2)} & \bullet & P_j^{(i)}
\end{array} \right)$$

On retrouve les mêmes matrices empilées le long de la diagonale $d^{(i)}$.

$$H_\infty = \begin{pmatrix} \cdots & \boxed{c_i} & \boxed{c_{i+1}} & \boxed{c_{i+2}} & \cdots & \boxed{c_{i+j}} & \cdots \\ \cdots & c_{i+1} & c_{i+2} & c_{i+3} & \cdots & c_{i+j+1} & \cdots \\ \cdots & c_{i+2} & c_{i+3} & c_{i+4} & \cdots & c_{i+j+2} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \cdots & c_{i+j} & c_{i+j+1} & c_{i+j+2} & \cdots & c_{i+2j} & \cdots \end{pmatrix}.$$

Les matrices encadrées permettent de caractériser des polynômes orthogonaux adjacents par rapport à la fonctionnelle renversée $\tilde{c} = \{c_{i+2j}, \dots, c_{i+1}, c_i\}$. On peut donc calculer récursivement ces polynômes sans l'aide de la fonctionnelle si on dispose des deux derniers. En les renversant, la propriété (1) nous confirme que nous retrouvons les polynômes cherchés.

Le déplacement que l'on vient d'effectuer le long d'une diagonale montante utilise au préalable deux éléments consécutifs sur cette diagonale. On ne peut pas se servir de ce raisonnement pour les diagonales montantes passant par $P_{j+1}^{(i)}$ ou $P_j^{(i+1)}$ parce que l'on ne dispose initialement que d'un élément sur ces diagonales (cf tableau (2.16)). Pour y remédier on exploite encore le renversement. On souhaite obtenir une relation $P_{j+1}^{(i)}, P_j^{(i+1)} \longrightarrow P_j^{(i+2)}$. Détaillons les polynômes orthogonaux qui leur correspondent pour la fonctionnelle renversée \tilde{c} . En notant $R_j^{(i)}$ le polynôme de degré j orthogonal par rapport à la fonctionnelle $\tilde{c}^{(i)}$, correspondent alors les polynômes suivants

$$\begin{cases} P_{j+1}^{(i)} & \leftrightarrow R_{j+1}^{(1)} \\ P_j^{(i+1)} & \leftrightarrow R_j^{(2)} \\ P_j^{(i+2)} & \leftrightarrow R_j^{(1)} \end{cases} \quad (2.17)$$

Il suffit alors de remarquer que la relation $R_{j+1}^{(1)}, R_j^{(2)} \longrightarrow R_j^{(1)}$ découle de (2.15) et qu'elle n'emploie pas les moments. On peut donc calculer les polynômes des deux diagonales montantes manquantes sans faire appel aux moments.

A partir de là, les moments dont on a besoin pour obtenir de nouveaux polynômes dépendent des directions de déplacement. En ce qui concerne la diagonale qui borde inférieurement le triangle précédent, on doit faire appel à la relation $P_{j-1}^{(i)}, P_j^{(i)} \longrightarrow P_{j-1}^{(i-1)}$ et celle-ci utilise les j premiers moments de la fonctionnelle. En effet, $P_{j-1}^{(i)}$ et $P_j^{(i)}$ vérifient les conditions d'orthogonalité

$$c^{(i-1)}(x^s P_{j-1}^{(i)}) = 0 \text{ pour } s = 1, \dots, j-1$$

et

$$c^{(i-1)}(x^s P_j^{(i)}) = 0 \text{ pour } s = 1, \dots, j.$$

Donc $P_{j-1}^{(i-1)}$ est la combinaison linéaire qui assure la première condition d'orthogonalité par rapport à $c^{(i-1)}$. Le degré de la combinaison étant $j - 1$ la condition ne nécessite que les j premiers moments.

Au contraire, si l'on souhaite calculer des polynômes le long de la diagonale montante qui passe par $P_j^{(i)}$, il faudra utiliser la dernière moitié des moments de la matrice qui les caractérise. Pour conclure ce paragraphe, le tableau suivant représente en caractères gras tous les polynômes que l'on peut obtenir à partir de deux éléments consécutifs d'une diagonale descendante. Nous indiquons en bordure, la partie de la fonctionnelle qu'il est nécessaire de connaître pour en calculer d'autres. On symbolise par un (d) les situations qui nécessitent l'intervention des moments de la fonctionnelle qui bordent le système par la gauche et par un (f) celles qui au contraire utilisent le bordage à droite.

$$\begin{array}{cccccccccccc}
 P_0 & P_0^{(1)} & \dots & (d) & P_0^{(i)} & \dots & \dots & P_0^{(i+j+1)} & \dots & \dots & P_0^{(i+2j+2)} & (f) & \dots \\
 & P_1 & \ddots & & \ddots & \ddots & & \vdots & & & & & \\
 & & \ddots & \ddots & & (d) & P_j^{(i)} & P_j^{(i+1)} & P_j^{(i+2)} & (f) & & & \\
 & & & \ddots & \ddots & (d) & P_{j+1}^{(i)} & (f) & & & & & \\
 & & & & \ddots & \ddots & & & \ddots & & & &
 \end{array} \tag{2.18}$$

Les coefficients des relations de récurrences peuvent être utilisés à d'autres fins que le calcul de polynômes orthogonaux et dans bien des situations la connaissance des coefficients de la récurrence est plus utile si ce n'est plus stable que les coefficients des polynômes eux-mêmes. Nous abordons avec la matrice de Jacobi une autre forme de représentation des polynômes orthogonaux.

2.5 La matrice de Jacobi

Si, réciproquement, on se donne une famille de polynômes satisfaisant une relation de récurrence à trois termes, qu'en est-il de l'orthogonalité? Le théorème de Favard ([45]), énonce que toute famille de polynômes $\{P_i\}$ qui obéit à une relation de récurrence (1.7), avec $A_{k+1} \in]0, \infty[$, $B_{k+1} \in \mathbb{R}$ et $C_{k+1} = A_k$, est orthogonale par rapport à une mesure positive. Ce théorème a été généralisé au cas d'une fonctionnelle quelconque dans ([39]). Non seulement la récurrence caractérise ainsi l'orthogonalité, mais elle donne en outre, au travers de ses coefficients, des informations sur les polynômes orthogonaux. C'est le cas par exemple de la propriété d'entrelacement de leurs racines ([100]). La matrice tridiagonale constituée par les coefficients de la récurrence (1.8) est appelée matrice de

Jacobi. Elle est donnée à l'ordre n par

$$J_n = \begin{pmatrix} -\alpha_1 & 1 & 0 & \cdots & 0 \\ -\beta_2 & -\alpha_2 & 1 & \ddots & \vdots \\ 0 & -\beta_3 & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & 1 \\ 0 & \cdots & 0 & -\beta_{n+1} & -\alpha_{n+1} \end{pmatrix}. \quad (2.19)$$

En notant I_n la matrice identité d'ordre n , J_n vérifie la propriété

$$P_{n+1}(x) = \det(xI_n - J_n).$$

P_{n+1} apparaît comme le polynôme caractéristique de J_n et ses racines comme les valeurs propres de J_n . Pour le calcul des racines des polynômes orthogonaux, on peut utiliser de nombreux procédés, éventuellement complémentaires. En partant de J_n , une succession de factorisations LU et UL forme l'algorithme qd ([94]) et permet une bonne localisation des racines. En suivant le même principe, on obtient l'algorithme QR ([58], p. 420) qui conserve aussi la structure tridiagonale. Enfin, dès qu'une bonne approximation d'une racine est disponible, une méthode de type Newton ([88], p. 254) assure une convergence quadratique. La récurrence offre par ailleurs bien d'autres intérêts pratiques.

Chapitre 3

L'algorithme régressif

3.1 Introduction

Plusieurs raisons nous ont motivé à réaliser un algorithme de calcul régressif pour les polynômes orthogonaux. La première d'entre elles était d'approfondir les liens entre la récurrence à trois termes et l'algorithme d'Euclide et de là, construire chacun d'eux comme une alternative à l'autre. Que pouvait-on espérer d'un changement de sens dans une récurrence polynômiale? Tout d'abord, la possibilité de renverser la récurrence laissait envisager un transfert de méthodologie en appliquant les connaissances acquises sur l'algorithme d'Euclide au domaine des polynômes orthogonaux, notamment les méthodes modulaires et p -adiques. Par la suite se sont accumulées d'autres intentions, comme celle de sauter par-dessus les étapes numériquement mal conditionnées au cours des récurrences. Enfin, il s'est avéré au cours des recherches bibliographiques que certains travaux allaient dans ce sens. L'un des plus significatifs étant le calcul du pgcd de deux polynômes par la résolution d'un système de Hankel ([96]).

Ce chapitre expose différents aspects algébriques qui sont apparus au cours de l'élaboration de l'algorithme régressif. Le premier paragraphe est consacré à la liaison entre la récurrence à trois termes et l'algorithme d'Euclide. Nous avons regroupé au second paragraphe les rappels concernant la transformée de Fourier discrète et son rôle croissant dans bon nombre de méthodes polynômiales. Au paragraphe suivant, nous illustrons l'importance de la transformée de Fourier discrète au travers de deux problèmes. Le quatrième paragraphe est consacré à la construction de l'algorithme régressif dont le dernier paragraphe décrit l'algorithme et un exemple dans la situation des corps finis.

3.2 L'algorithme d'Euclide et la relation de récurrence à trois termes

Les conditions imposées aux degrés respectifs de P_{k+1} , P_k et P_{k-1} permettent de faire apparaître dans (1.8) le polynôme $\beta_{k+1}P_{k-1}$ comme le reste de la division Euclidienne de P_{k+1} par P_k . Pour éclairer ce constat, on peut mettre en parallèle la représentation

utilisant la matrice de Jacobi avec la méthode de Givens pour le calcul des valeurs propres d'une matrice tridiagonale symétrique et l'algorithme de Sturm sous-jacent ([35],[9]). L'algorithme d'Euclide et la récurrence à trois termes sont donc deux facettes d'un même algorithme.

La généralisation au cas non régulier est étudiée en détails dans ([39], p.88), où l'on trouve le théorème suivant. On note $P_{pr(k)}$ le polynôme régulier de plus haut degré qui précède le k -ième polynôme régulier P_k .

Théorème 1

Soit P_k et $P_{pr(k)}$ deux polynômes unitaires premiers entre eux. Il existe une suite de moments c_j , pour $j \in \mathbb{N}$, $0 \leq j \leq 2k - 1$, unique dès qu'on a fixé le premier moment non nul à partir de c_0 , telle que les polynômes P_k et $P_{pr(k)}$, ainsi que tous les polynômes P_i formés par la suite des restes déduits de P_k et $P_{pr(k)}$ par application de l'algorithme d'Euclide, soient orthogonaux par rapport à toute fonctionnelle linéaire c dont les $2k + 2$ premiers moments sont les c_j .

Il s'ensuit que les polynômes réguliers successifs P_i et $P_{pr(i)}$ sont eux aussi premiers entre eux dès que $P_0 \neq 0$. Il est important de noter que les premiers moments peuvent être nuls, ce que montre l'exemple qui suit.

Exemple 8

Construisons une fonctionnelle à partir de deux polynômes P_4 et P_3 , de degré 4 et 3 respectivement, premiers entre eux et dont le reste obtenu dans la division Euclidienne de P_4 par P_3 est de degré zéro. On choisit par exemple

$$P_4(x) = x^4 + x^3 + x^2 + x + 1$$

et

$$P_3(x) = x^3 + x + .$$

Si on note (c_0, c_1, \dots) les moments de la fonctionnelle c par rapport à laquelle ils sont orthogonaux, les conditions d'orthogonalité imposent les relations suivantes.

$$\begin{aligned} c(P_3) = 0 &\Leftrightarrow c_3 + c_1 = 0 \\ c(xP_3) = 0 &\Leftrightarrow c_4 + c_2 = 0 \\ c(x^2P_3) = 0 &\Leftrightarrow c_5 + c_3 = 0 \end{aligned}$$

$$\begin{aligned} c(P_4) = 0 &\Leftrightarrow c_4 + c_3 + c_2 + c_1 + c_0 = 0 \\ c(xP_4) = 0 &\Leftrightarrow c_5 + c_4 + c_3 + c_2 + c_1 = 0 \\ c(x^2P_4) = 0 &\Leftrightarrow c_6 + c_5 + c_4 + c_3 + c_2 = 0 \\ c(x^3P_4) = 0 &\Leftrightarrow c_7 + c_6 + c_5 + c_4 + c_3 = 0 \end{aligned}$$

Pour que la fonctionnelle ne soit pas irrégulière on est amené à poser $c(x^3 P_3) \neq 0$ et $c(x^4 P_4) \neq 0$. La résolution des équations conduit d'une part à

$$c_0 = c_1 = c_3 = c_5 = c_6 = 0$$

et d'autre part à

$$c_4 \neq 0, \quad c_4 = -c_2 = -c_7, \quad c_8 \neq 0.$$

En posant $c_4 = 1$ et $c_8 = 1$ la fonctionnelle c devient

$$c = (0, 0, -1, 0, 1, 0, 0, -1, 1, \dots).$$

Le nombre de zéros présents au début de la fonctionnelle dépend ici directement du saut en degré effectué par la division Euclidienne de P_4 par P_3 ,

$$P_4(x) = (x + 1)P_3(x) + 1.$$

◇

Remarquons que cette situation peut être problématique en mode de calcul formel pour initialiser la récurrence à trois termes. De même en mode de calcul de précision finie, la difficulté peut provenir de premiers moments de très petites tailles.

Reprenons le cas de forte régularité. D'après le théorème de Bezout, il existe alors une famille de polynômes S_i qui satisfait les conditions suivantes

$$P_{i-1}S_i - P_iS_{i-1} = c(P_{i-1}P_{i-1}). \quad (3.1)$$

Les polynômes P_i vérifient une relation de récurrence à trois termes qui n'est autre que (1.7) normalisée par la condition $B_{i+1} = 1$. Les polynômes $(-1)^i S_i$ suivent la même relation de récurrence que les polynômes P_i , mais avec des conditions initiales modifiées, $S_{-1} = 1$ et $S_0 = 0$. Dans le cas de forte régularité, une telle récurrence à trois termes, sans conditions initiales, admet pour ensemble de solutions un sous-espace vectoriel de dimension deux de l'espace vectoriel des suites de polynômes. En fixant arbitrairement $P_{-1} = 0$, les familles solutions S_i et P_i sont linéairement indépendantes et fournissent une base de l'espace des solutions. Cette propriété (3.1) se généralise au cas des fonctionnelles non fortement régulières et les S_i peuvent se calculer directement à partir des P_i et de la fonctionnelle.

3.2.1 Définition

Les polynômes S_i sont dits associés aux P_i ; on les obtient également par la formule suivante

$$c \left(\frac{P_i(t) - P_i(x)}{t - x} \right) = S_i(x), \quad \forall i \in \mathbb{N}, \quad (3.2)$$

où la fonctionnelle agit sur la variable t . On note $f(x) = \sum c_i x^i$ la série associée à c . Les polynômes S_i sont orthogonaux par rapport à la fonctionnelle adjacente à d notée $d^{(2)}$ en respectant les notations introduites en préambule, où d est la fonctionnelle associée à la série $g(x)$, dite série réciproque de $f(x)$ et qui vérifie $g(x)f(x) = 1$ (voir [14]).

3.2.2 Calcul des polynômes associés

Représentons l'opération (3.2) sous forme matricielle. Soit $S_{n-1}(t) = \sum_{i=0}^{i=n-1} s_i t^i$ le polynôme de degré $n-1$ associé à $P_n(t) = \sum_{i=0}^{i=n} p_i t^i$ de degré n pour la fonctionnelle c .

$$\begin{aligned} S_{n-1}(x) &= c \left(\frac{P_n(t) - P_n(x)}{t - x} \right) \\ &= c \left(\frac{\sum_{i=0}^{i=n} p_i t^i - \sum_{i=0}^{i=n} p_i x^i}{t - x} \right) \\ &= c \left(\sum_{i=1}^{i=n} p_i \frac{t^i - x^i}{t - x} \right) \\ &= c \left(\sum_{i=1}^{i=n} p_i \left(\sum_{j=0}^{j=i-1} x^j t^{i-1-j} \right) \right). \end{aligned}$$

Les indices parcourent l'ensemble $\{(i, j), 1 \leq i \leq n, 0 \leq j \leq i-1\}$, que l'on peut également décrire comme $\{(i, j), 0 \leq j \leq n-1, j+1 \leq i \leq n\}$. En appliquant la fonctionnelle sur la variable t la somme prend alors la forme

$$= \sum_{j=0}^{j=n-1} x^j \left(\sum_{i=j+1}^{i=n} p_i c_{i-1-j} \right).$$

On en déduit que $s_j = \sum_{i=j+1}^{i=n} p_i c_{i-1-j}$. Les coefficients s_i sont donc obtenus par le produit matriciel suivant

$$\begin{pmatrix} c_0 & c_1 & \cdots & c_{n-2} & c_{n-1} \\ 0 & c_0 & \cdots & c_{n-3} & c_{n-2} \\ \vdots & 0 & & \vdots & \vdots \\ 0 & \vdots & & c_0 & c_1 \\ 0 & 0 & \cdots & 0 & c_0 \end{pmatrix} \begin{pmatrix} p_1 \\ p_2 \\ \vdots \\ \vdots \\ p_n \end{pmatrix} = \begin{pmatrix} s_0 \\ s_1 \\ \vdots \\ \vdots \\ s_{n-1} \end{pmatrix}. \quad (3.3)$$

3.2.3 L'identité de Christoffel-Darboux

Une famille $\{P_k\}$ de polynômes unitaires orthogonaux par rapport à une fonctionnelle linéaire c satisfait une relation de Christoffel-Darboux ([15]),

$$(x-t) \sum_{i=0}^{i=n} \frac{1}{h_i} P_i(x) P_i(t) = \frac{1}{h_n} [P_{n+1}(x) P_n(t) - P_{n+1}(t) P_n(x)], \quad (3.4)$$

avec $h_i = c(P_i^2)$.

Pour la démonstration, on procède par récurrence sur n . La relation à trois termes permet de transmettre la propriété d'un rang sur le suivant. Cette formule sert par exemple pour la construction de représentations conformes ([69]). La notion qui lui est

associée est celle de noyau reproduisant très utile pour la Physique Mathématique. Le noyau reproduisant est la fonction définie par

$$K_n(x, t) = \sum_{i=0}^{i=n} \frac{1}{h_i} P_i(x) P_i(t)$$

Cette fonction a la propriété de reconstituer le polynôme de départ comme l'exprime la formule suivante,

$$c(Q(x)K_n(x, t)) = Q(y) \text{ pour tout polynôme } Q \text{ de degré inférieur à } n.$$

Un résultat récent obtenu par C.Brezinski ([17]) démontre que pour une famille de polynômes, il y a équivalence entre la relation de récurrence à trois termes et la formule de Christoffel-Darboux. On pourrait donc présenter les polynômes orthogonaux formels à partir des noyaux reproduisants qui vérifient (3.4). Les généralisations de (3.4) fournissent différents cadres auxquels on peut étendre les outils de la théorie des polynômes orthogonaux. On trouvera par exemple la généralisation au cas non régulier dans ([39]) et l'extension de la formule aux familles de polynômes adjacents dans ([19]). On peut citer également la généralisation aux polynômes biorthogonaux définis par une matrice de moments de type Toeplitz développée dans ([31]), ou encore le cas des matrices polynômiales orthogonales relativement à une matrice de mesures rectangulaire ([6]). Outre son utilisation dans la démonstration de l'entrelacement des racines des polynômes orthogonaux, on l'emploie en algèbre linéaire au travers des matrices Bezoutiennes ([105],[67]). C'est ce dernier aspect que nous mettrons en œuvre au paragraphe (6.1) pour observer l'inverse d'une matrice de Hankel.

3.3 Transformée de Fourier discrète (DFT) et convolution cyclique

3.3.1 Définition

L'opérateur de transformée de Fourier discrète (DFT en abrégé) d'ordre n peut être considéré comme un isomorphisme de l'espace vectoriel \mathbb{C}^n dont la matrice associée dans la base canonique est la matrice de Fourier

$$F_w = \begin{pmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & w & w^2 & \dots & w^{n-1} \\ 1 & w^2 & w^4 & \dots & w^{2(n-1)} \\ \vdots & & & & \\ 1 & w^{n-1} & w^{2(n-1)} & \dots & w^{(n-1)^2} \end{pmatrix} \quad (3.5)$$

avec $w = e^{-2i\pi/n}$.

Si $V^T = (v_0, v_1, \dots, v_{n-1})$ est un vecteur de \mathbb{C}^n , la DFT de V est le vecteur $\hat{V}^T = (\hat{v}_0, \hat{v}_1, \dots, \hat{v}_{n-1})$ défini par

$$\hat{v}_k = \sum_{s=0}^{n-1} v_s e^{-2i\pi sk/n}.$$

3.3.2 Transformée inverse

F_w est symétrique. De plus, les vecteurs colonnes de $\frac{1}{\sqrt{n}}F_w$ forment une base orthonormale de \mathbb{C}^n . F_w est donc unitaire à une constante de normalisation près. On obtient alors, en notant F_w^* l'adjoint de F_w et I_n la matrice identité de \mathbb{C}^n , l'égalité suivante $F_w^*F_w = nI_n$. Il suffit alors de rappeler que, pour une racine de l'unité, $\bar{w} = w^{-1}$ pour pouvoir conclure que

$$F_w^{-1} = \frac{1}{n}F_{w^{-1}}. \quad (3.6)$$

3.3.3 Intérêt algorithmique de la transformée de Fourier rapide

La matrice F_w possède des propriétés de symétrie et de périodicité qui ont permis de réduire notablement le nombre d'opérations nécessaires pour effectuer son produit sur un vecteur. On pourra trouver des algorithmes de transformée de Fourier rapide (FFT) dans ([34]) et ([88]). Pour un ordre n égal à une puissance de 2 on passe ainsi d'un coût en $O(n^2)$ à un nombre d'opérations en $O(n \log_2(n))$. En prenant par exemple $n = 32$, on gagne déjà un facteur 20. Lorsque n admet une décomposition en produits d'entiers de la forme $n = pq$, la DFT peut être calculée avec un nombre de multiplications proportionnel à $n(p+q)$. Dans la pratique, il faut augmenter le nombre de données artificiellement pour se ramener en $n = 2^k$. Dans le cas où n admet un petit facteur différent de 2, on peut utiliser en complément la méthode de Winograd ([107]).

3.3.4 Evaluation et interpolation polynomiale

La matrice F_w d'ordre n est également une matrice de type Vandermonde. Plus précisément, il s'agit d'une matrice d'interpolation sur les racines d'ordre n de l'unité w^s , $s = 0, \dots, n-1$. Si maintenant on associe au vecteur $V^T = (v_0, v_1, \dots, v_{n-1})$ le polynôme $V_n(x) = \sum_{i=0}^{n-1} v_i x^i$, alors, calculer $U = F_w V$ revient à évaluer $V_n(x)$ aux racines d'ordre n de l'unité, c'est-à-dire $u_i = V_n(w^i)$ pour $i = 0, \dots, n-1$. Ce point de vue est particulièrement approprié pour décrire la liaison établie par la DFT entre les notions de produit ponctuel et de produit de convolution circulaire.

3.3.5 Convolution circulaire

La convolution circulaire de deux vecteurs C et V de \mathbb{C}^n de composantes $(c_i)_{0 \leq i \leq n-1}$ et $(v_i)_{1 \leq i \leq n}$ respectivement, est le vecteur $C * V$ défini par

$$C * V = \begin{pmatrix} c_0 & c_{n-1} & \cdots & c_2 & c_1 \\ c_1 & c_0 & \ddots & \ddots & c_2 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ c_{n-2} & \ddots & \ddots & c_0 & c_{n-1} \\ c_{n-1} & c_{n-2} & \cdots & c_1 & c_0 \end{pmatrix} \begin{pmatrix} v_0 \\ v_1 \\ \vdots \\ \vdots \\ v_{n-1} \end{pmatrix}. \quad (3.7)$$

L'opération $*$ entre C et V sous-entend que le vecteur C est transformé en la matrice carrée cyclique précédente. Cette opération est commutative. Si nous renversons les composantes du vecteur C nous obtenons le vecteur $\tilde{C} = (c_{n-1}, c_{n-2}, \dots, c_1, c_0)$ qui vérifie

$$\tilde{C} * V = \begin{pmatrix} c_{n-1} & c_0 & \cdots & c_{n-3} & c_{n-2} \\ c_{n-2} & c_{n-1} & \ddots & \ddots & c_{n-3} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ c_1 & \ddots & \ddots & c_{n-1} & c_0 \\ c_0 & c_1 & \cdots & c_{n-2} & c_{n-1} \end{pmatrix} \begin{pmatrix} v_0 \\ v_1 \\ \vdots \\ \vdots \\ v_{n-1} \end{pmatrix}. \quad (3.8)$$

3.3.6 Convolution et produit ponctuel

La convolution circulaire coïncide avec le produit des polynômes dans un contexte bien précis. Notons $C_n(x)$, $V_n(x)$ et $(C * V)_n(x)$ les polynômes associés comme précédemment aux vecteurs C , V et $(C * V)$ et imposons à l'indéterminée x la contrainte $x^n = 1$. Il en résulte l'égalité suivante

$$(C * V)_n(x) = C_n(x).V_n(x). \quad (3.9)$$

Remarquons qu'en évaluant ces deux polynômes aux racines d'ordre n de l'unité qui vérifient la contrainte $x^n = 1$, on obtient

$$F_w(C * V) = F_w(C).F_w(V), \quad (3.10)$$

où le produit des vecteurs du membre de droite doit être considéré terme à terme. La DFT transforme ainsi un produit de convolution circulaire en un produit ponctuel.

3.3.7 Produit de polynômes

Le produit de deux polynômes peut s'effectuer rapidement par l'emploi de la FFT. Soit p et q deux polynômes dont les degrés sont inférieurs ou égaux à $n - 1$. Pour évaluer p et q aux racines primitives de l'unité d'ordre $2n$ $1, w, \dots, w^i, \dots, w^{2n-1}$, il suffit d'appliquer deux FFT. Les $2n$ produits $p(w^i)q(w^i)$ fournissent les valeurs du polynôme produit aux points précédents. Une DFT inverse réalise l'interpolation sur les points $1, w, \dots, w^i, \dots, w^{2n-1}$, ce qui requiert une FFT et $2n$ divisions. Le procédé utilise donc un total de $O(n \log(n))$ opérations arithmétiques à comparer au produit usuel en $O(n^2)$. Néanmoins, il convient d'être prudent lors de l'utilisation de ce procédé pour le calcul de produits de polynômes de petit degré; en effet les coefficients négligés dans le calcul de la complexité deviennent

alors prépondérants. D'autre part, l'emploi de cette technique nécessite une sévère mise en garde quant à l'instabilité numérique qu'elle génère dans certains algorithmes. Citons à titre d'avertissement un extrait de ([2], p. 105) sur la question des algorithmes "superfast" pour le calcul des approximants de Padé et qui reposent sur cette procédure: << Methods of this kind are remarkably elegant, but because they are so prone to failure we do not elaborate the details here...strong caveats on their usage are expressed by Ammar and Gragg ([1]) and Bunch ([27])...>>. Par contre, cette restriction disparaît lorsque l'on emploie le calcul formel.

Si l'on représente un polynôme par le vecteur des coefficients des monômes qui le composent, alors le produit $s(x) = p(x)q(x) = s_0 + s_1x + \dots + s_{m+n}x^{n+m}$ peut être décrit par un produit matrice vecteur de la forme

$$\begin{pmatrix} p_0 & 0 & \dots & \dots & 0 \\ p_1 & p_0 & \ddots & & \vdots \\ \vdots & p_1 & \ddots & \ddots & \vdots \\ \vdots & & \ddots & p_0 & 0 \\ p_{n-1} & & & p_1 & p_0 \\ p_n & p_{n-1} & & & p_1 \\ 0 & p_n & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & p_n & p_{n-1} \\ 0 & \dots & \dots & 0 & p_n \end{pmatrix} \begin{pmatrix} p_0 \\ p_1 \\ \vdots \\ \vdots \\ p_m \end{pmatrix} = \begin{pmatrix} s_0 \\ s_1 \\ \vdots \\ \vdots \\ s_{n-1} \\ s_n \\ \vdots \\ s_{n+m-2} \\ s_{n+m-1} \\ s_{n+m} \end{pmatrix}, \quad (3.11)$$

où $p(x) = p_0 + p_1x + \dots + p_nx^n$ et $q(x) = q_0 + q_1x + \dots + q_mx^m$. Cette dernière remarque est, comme nous allons le constater, la base de bon nombre de liaisons entre les différents aspects des polynômes qui jalonnent ce travail.

3.4 Différentes connexions entre les matrices de Toeplitz, les matrices de Hankel et la DFT

3.4.1 Produit d'une matrice de Toeplitz sur un vecteur

$T = [t_{i,j}]$ est une matrice de Toeplitz si ses coefficients vérifient $t_{i,j} = t_{i+k,j+k}$ pour tout $k \geq 0$, c'est-à-dire avec des coefficients égaux le long des diagonales descendantes. Le produit d'une matrice de Toeplitz sur un vecteur s'apparente au produit de polynômes; ce qui permet d'utiliser la FFT pour réduire la complexité de ce produit. Soit $W_{3n-3}(x) = \sum_{i=0}^{3n-3} w_i x^i$ le polynôme produit des polynômes $C_{2n-2}(x) = \sum_{i=0}^{2n-2} c_i x^i$ et $D_{n-1}(x) =$

$\sum_{i=0}^{n-1} d_i x^i$, en utilisant une écriture matricielle de l'opération on obtient

$$\begin{pmatrix}
 c_0 & 0 & \cdots & \cdots & 0 \\
 c_1 & c_0 & \ddots & & \vdots \\
 \vdots & \ddots & \ddots & \ddots & \vdots \\
 \vdots & & & & 0 \\
 \hline
 c_{n-1} & c_{n-2} & & c_1 & c_0 \\
 c_n & c_{n-1} & \ddots & & c_1 \\
 \vdots & \ddots & \ddots & \ddots & \vdots \\
 c_{2n-3} & & & c_{n-1} & c_{n-2} \\
 c_{2n-2} & c_{2n-3} & & c_n & c_{n-1} \\
 \hline
 0 & & & & c_n \\
 \vdots & \ddots & \ddots & & \vdots \\
 & & & c_{2n-2} & c_{2n-3} \\
 0 & \cdots & \cdots & 0 & c_{2n-2}
 \end{pmatrix}
 \begin{pmatrix}
 d_0 \\
 d_1 \\
 \vdots \\
 d_{n-1}
 \end{pmatrix}
 =
 \begin{pmatrix}
 p_0 \\
 p_1 \\
 \vdots \\
 \vdots \\
 p_{n-1} \\
 p_n \\
 \vdots \\
 p_{2n-3} \\
 p_{2n-2} \\
 \vdots \\
 \vdots \\
 p_{3n-4} \\
 p_{3n-3}
 \end{pmatrix}$$

(3.12)

Dans la partie encadrée, est mis en évidence le produit d'une matrice de Toeplitz par un vecteur. En sélectionnant les composantes appropriées du produit des deux polyômes, on aura évalué le produit recherché et ainsi ramené la complexité de $O(n^2)$ à $O(n \log(n))$. Cette propriété des matrices de Toeplitz ou de Hankel est fondamentale en traitement du signal lorsqu'il s'agit d'évaluer fréquemment un produit de matrice vecteur. Par exemple, dans la résolution d'un système linéaire de type Toeplitz, une méthode de type gradient conjugué, éventuellement précédée d'un préconditionnement par une matrice circulante, chaque étape de l'algorithme nécessite le produit matrice vecteur.

3.4.2 Inversion d'une matrice de Toeplitz triangulaire inférieure

L'inversion d'une matrice de Toeplitz triangulaire inférieure peut aussi tirer parti de l'accélération du calcul du produit de polynômes. Une matrice de Toeplitz triangulaire inférieure T est entièrement déterminée par sa première colonne (t_0, \dots, t_{n-1}) . De plus, l'inverse d'une telle matrice est aussi de type Toeplitz triangulaire inférieure. En effet, déterminer la première colonne de la matrice T^{-1} revient à déterminer le polyôme $Q_0(x)$ tel que $T(x)Q_0(x) = 1 \pmod{x^n}$ en se référant à la représentation matricielle du produit. Or la $k+1$ ième colonne de T^{-1} revient à déterminer le polynôme $Q_k(x)$ tel que $T(x)Q_k(x) = x^k \pmod{x^n}$. Il suffit d'observer alors que $Q_k(x) = x^k Q_0(x)$ est une solution

qui convient pour en déduire, par unicité de l'inverse, que la solution cherchée est une matrice de Toeplitz triangulaire inférieure et donc, entièrement déterminée par sa première colonne, c'est-à-dire par les coefficients de $Q_0(x)$. Pour trouver $Q(x)$ tel que $T(x)Q(x) = 1 \pmod{x^n}$, une stratégie de type diviser pour régner décompose les polynômes de la manière suivante pour n pair, $(T_1(x) + x^{\frac{n}{2}}T_2(x))(Q_1(x) + x^{\frac{n}{2}}Q_2(x)) = 1 \pmod{x^n}$, où $T_1(x), T_2(x), Q_1(x)$ et $Q_2(x)$ sont des polynômes de degré $n/2$. En développant, il apparaît que $T_1(x)Q_1(x) + x^{\frac{n}{2}}(Q_2(x)T_1(x) + T_2(x)Q_1(x)) = 1 \pmod{x^n}$. c'est-à-dire, dans un premier temps

$$\begin{aligned} T_1(x)Q_1(x) + x^{\frac{n}{2}}(Q_2(x)T_1(x) + T_2(x)Q_1(x)) &= 1 \pmod{x^{n/2}} \\ \Leftrightarrow T_1(x)Q_1(x) &= 1 \pmod{x^{n/2}} \end{aligned}$$

puis, après simplification

$$\begin{aligned} Q_2(x)T_1(x) + T_2(x)Q_1(x) &= 0 \pmod{x^{n/2}} \\ \Leftrightarrow Q_2(x)T_1(x) &= -T_2(x)Q_1(x) \pmod{x^{n/2}}. \end{aligned}$$

Ce qui nous amène au système suivant où les degrés sont ramenés à $n/2$

$$\begin{aligned} T_1(x)Q_1(x) &= 1 \pmod{x^{n/2}} \\ T_1(x)Q_2(x) &= -T_2(x)Q_1(x) \pmod{x^{n/2}}. \end{aligned}$$

Observons que la solution de la première équation du système donne la solution de la seconde équation. On obtient par multiplication

$$-T_1(x)Q_1(x)T_2(x)Q_1(x) = -T_2(x)Q_1(x) \pmod{x^{n/2}},$$

et, si l'on identifie avec la deuxième équation,

$$Q_2(x) = -Q_1(x)T_2(x)Q_1(x) \pmod{x^{n/2}}.$$

3.4.3 Résolution d'un système de Toeplitz Hessenberg inférieur

En combinant les méthodes précédentes, on peut présenter un algorithme en $O(n \log(n))$ pour résoudre des systèmes de Toeplitz Hessenberg inférieurs. Un système de Toeplitz Hessenberg inférieur est de la forme

$$\begin{pmatrix} c_{n-1} & c_n & 0 & \cdots & 0 \\ c_{n-2} & c_{n-1} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ c_1 & \ddots & \ddots & c_{n-1} & c_n \\ c_0 & c_1 & \cdots & c_{n-2} & c_{n-1} \end{pmatrix} \begin{pmatrix} X \\ \vdots \\ X \end{pmatrix} = \begin{pmatrix} b_0 \\ b_1 \\ \vdots \\ b_{n-1} \end{pmatrix}. \quad (3.13)$$

On complète le système précédent pour obtenir un système de matrice T Toeplitz triangulaire inférieure de la manière suivante

$$\begin{pmatrix} c_n & 0 & \cdots & & 0 \\ c_{n-1} & c_n & \ddots & & \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ c_1 & \ddots & \ddots & c_n & 0 \\ c_0 & c_1 & \cdots & c_{n-1} & c_n \end{pmatrix} \begin{pmatrix} X \\ \\ \\ 0 \end{pmatrix} = \begin{pmatrix} c_n x_0 \\ b_0 \\ b_1 \\ \vdots \\ b_{n-1} \end{pmatrix}. \quad (3.14)$$

On peut dès lors calculer T^{-1} pour l'appliquer sur le second membre

$$\begin{pmatrix} X \\ \\ \\ 0 \end{pmatrix} = T^{-1} \begin{pmatrix} c_n x_0 \\ b_0 \\ b_1 \\ \vdots \\ b_{n-1} \end{pmatrix} = x_0 T^{-1} \begin{pmatrix} c_n \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} + T^{-1} \begin{pmatrix} 0 \\ b_0 \\ b_1 \\ \vdots \\ b_{n-1} \end{pmatrix}. \quad (3.15)$$

La dernière équation détermine l'inconnue x_0 . Si on note $U^T = (u_0, \dots, u_n)$ la dernière ligne de la matrice T^{-1} , on a

$$x_0 = \frac{-\sum_{i=1}^{i=n-1} u_i b_i}{u_0 c_n}.$$

On en déduit aisément X par (3.15). Par une technique de bordage nous nous sommes ramenés à un problème pour lequel la méthode de la division pour régner s'applique.

3.4.4 Les matrices de Hankel et la convolution circulaire

Matrices de Hankel.

Une matrice de Hankel $n \times n$ est définie à partir de $2n - 1$ nombres complexes c_i , $i = 0, \dots, 2n - 2$ de la manière suivante

$$H_n = \begin{pmatrix} c_0 & c_1 & \cdots & c_{n-2} & c_{n-1} \\ c_1 & c_2 & \cdots & c_{n-1} & c_n \\ \vdots & \vdots & & \vdots & \vdots \\ c_{n-2} & c_{n-1} & \cdots & c_{2n-4} & c_{2n-3} \\ c_{n-1} & c_n & \cdots & c_{2n-3} & c_{2n-2} \end{pmatrix}.$$

Si on multiplie H_n par la matrice de permutation de ligne P_n , on obtient une matrice de Toeplitz

$$P_n H_n = \begin{pmatrix} 0 & 0 & \cdots & 0 & 1 \\ 0 & 0 & \cdots & 1 & 0 \\ \vdots & \vdots & & \vdots & \vdots \\ 0 & 1 & \cdots & 0 & 0 \\ 1 & 0 & \cdots & 0 & 0 \end{pmatrix} H_n = \begin{pmatrix} c_{n-1} & c_n & \cdots & c_{2n-3} & c_{2n-2} \\ c_{n-2} & c_{n-1} & \ddots & \ddots & c_{2n-3} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ c_1 & \ddots & \ddots & c_{n-1} & c_n \\ c_0 & c_1 & \cdots & c_{n-2} & c_{n-1} \end{pmatrix}. \quad (3.16)$$

Lien avec la convolution.

On peut clairement identifier $P_n H_n$ à la matrice de convolution dans (3.8) dès que l'on a

$$c_{n+i} = c_i, \text{ for } i = 0, \dots, n-1.$$

Pour tout vecteur Y de \mathbb{C}^n on a alors l'égalité

$$P_n H_n Y = C * Y, \tag{3.17}$$

avec $C = (c_{n-1}, c_{n-2}, \dots, c_1, c_0)$.

3.5 Polynômes orthogonaux relativement à une fonctionnelle linéaire cyclique et régulière

Les observations précédentes nous ont conduit à étudier les fonctionnelles linéaires dont la matrice des moments est de la forme d'une matrice de convolution. Dans ce paragraphe, nous ne considérons que des fonctionnelles tronquées.

Définition 2

Une fonctionnelle c de moments $(c_0, c_1, \dots, c_{2n-1})$ satisfaisants les conditions

$$c_{n+i} = c_i, \text{ pour } i = 0, \dots, n-1$$

sera appelée fonctionnelle cyclique.

3.5.1 Le dernier polynôme orthogonal relativement à une fonctionnelle cyclique

Propriété 2

Soit c une fonctionnelle régulière cyclique de moments (c_i) , $i \in \{0, \dots, 2n-1\}$ sur $\mathbb{C}_{2n-1}[x]$. Le polynôme unitaire de degré n orthogonal par rapport à c est

$$P_n(x) = x^n - 1.$$

preuve:

La régularité de c assure l'existence de P_n et l'unicité découle du choix unitaire. On

pose, $P_n(x) = \sum_{i=0}^{n-1} a_i x^i + x^n$, $C^T = (c_{n-1}, c_{n-2}, \dots, c_0)$ et $X^T = (a_0, a_1, \dots, a_{n-1})$. Les coefficients de P_n doivent satisfaire le système de Hankel (1.4)

$$\begin{pmatrix} c_0 & c_1 & \cdots & c_{n-2} & c_{n-1} \\ c_1 & c_2 & \cdots & c_{n-1} & c_0 \\ \vdots & \vdots & & \vdots & \vdots \\ c_{n-2} & c_{n-1} & \cdots & c_{n-4} & c_{n-3} \\ c_{n-1} & c_0 & \cdots & c_{n-3} & c_{n-2} \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ \vdots \\ a_{n-1} \end{pmatrix} = - \begin{pmatrix} c_0 \\ c_1 \\ \vdots \\ \vdots \\ c_{n-1} \end{pmatrix}. \quad (3.18)$$

La formule (3.16) appliquée au vecteur X cherché nous donne

$$P_n H_n X = -(c_{n-1}, c_{n-2}, \dots, c_0)^T.$$

Si on adopte la notation utilisée pour la convolution cyclique (3.7), on obtient

$$C * X = -C.$$

Appliquons la DFT à chaque membre de l'égalité précédente puis la formule (3.10)

$$\hat{C} \cdot \hat{X} = -\hat{C}$$

où \hat{C} et \hat{X} désignent les DFT respectives de C et X et le produit étant bien entendu terme à terme. Remarquons qu'une condition nécessaire pour la régularité de la fonctionnelle, qui apparaît clairement dans l'égalité précédente, consiste à imposer qu'aucune composante de \hat{C} ne soit nulle. On verra ultérieurement que cette condition est aussi suffisante. La solution du système résulte de $\hat{X} = -(1, 1, \dots, 1)^T$, puis de la formule d'inversion de la DFT (3.6), ce qui entraîne aussitôt que $X = -(1, 0, \dots, 0)^T$. Ce résultat peut également se déduire directement de la comparaison du second membre du système avec la première colonne de la matrice du système .

■

3.5.2 L'avant-dernier polynôme orthogonal

Nous allons à présent calculer l'avant-dernier polynôme orthogonal pour une fonctionnelle cyclique en nous ramenant par bordage à un système cyclique.

Propriété 3

Soit c une fonctionnelle cyclique régulière sur $\mathbb{C}_{2n-1}[x]$ et de moments (c_i) , $i \in \{0, \dots, 2n-1\}$. Le polynôme orthogonal unitaire de degré $n-1$ relativement à c est $P_{n-1}(x) = \sum_{i=0}^{n-2} a_i x^i + x^{n-1}$ où le vecteur $X^T = (a_0, a_1, \dots, a_{n-2})$ vérifie

$$\begin{pmatrix} X \\ 1 \end{pmatrix} = \alpha F^{-1} \left(\frac{\mathbf{1}}{FC} \right). \quad (3.19)$$

F désigne la matrice de Fourier (3.5) d'ordre n , $C^T = (c_{n-1}, c_{n-2}, \dots, c_0)$ et $\mathbf{1}^T = (1, 1, \dots, 1)$ de dimension n . α est alors une constante imposée par la normalisation unitaire de P_{n-1} .

preuve:

On pose $U^T = (c_{n-1}, c_0, c_1, \dots, c_{n-3})$. Les coefficients (a_i) sont obtenus par le système de Hankel suivant

$$\begin{pmatrix} c_0 & c_1 & \cdots & c_{n-3} & c_{n-2} \\ c_1 & c_2 & \cdots & c_{n-2} & c_{n-1} \\ \vdots & \vdots & & \vdots & c_0 \\ c_{n-3} & c_{n-2} & \cdots & & \vdots \\ c_{n-2} & c_{n-1} & c_0 & \cdots & c_{n-4} \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ \vdots \\ a_{n-2} \end{pmatrix} = - \begin{pmatrix} c_{n-1} \\ c_0 \\ c_1 \\ \vdots \\ c_{n-3} \end{pmatrix}.$$

En utilisant la méthode de bordage, on reconstitue un système de dimension n dont la matrice est circulante

$$\begin{pmatrix} \begin{pmatrix} c_0 & c_1 & \cdots & c_{n-3} & c_{n-2} \\ c_1 & c_2 & \cdots & c_{n-2} & c_{n-1} \\ \vdots & \vdots & & \vdots & c_0 \\ c_{n-3} & c_{n-2} & \cdots & & \vdots \\ c_{n-2} & c_{n-1} & c_0 & \cdots & c_{n-4} \end{pmatrix} & \begin{pmatrix} c_{n-1} \\ c_0 \\ c_1 \\ \vdots \\ c_{n-3} \\ c_{n-2} \end{pmatrix} \\ \begin{pmatrix} c_{n-1} & c_0 & c_1 & \cdots & c_{n-3} \end{pmatrix} & c_{n-2} \end{pmatrix} \begin{pmatrix} X \\ y \end{pmatrix} = \begin{pmatrix} - \begin{pmatrix} c_{n-1} \\ c_0 \\ c_1 \\ \vdots \\ c_{n-3} \end{pmatrix} + y \begin{pmatrix} c_{n-1} \\ c_0 \\ c_1 \\ \vdots \\ c_{n-3} \end{pmatrix} \\ U^t X + y c_{n-2} \end{pmatrix}.$$

On obtient donc à nouveau un problème de convolution comme dans l'équation (3.18) et il vient

$$P_n H_n \begin{pmatrix} X \\ y \end{pmatrix} = \begin{pmatrix} U^T X + c_{n-1} y \\ (y-1) P_{n-1} U \end{pmatrix}$$

alors

$$C * \begin{pmatrix} X \\ y \end{pmatrix} = \begin{pmatrix} U^T X + c_{n-1} y \\ (y-1) P_{n-1} U \end{pmatrix}.$$

y étant un nombre arbitraire, l'on peut donc choisir $y = 1$. En appliquant la transformée de Fourier et la formule (3.10), on a

$$FC \cdot F \begin{pmatrix} X \\ 1 \end{pmatrix} = \alpha F \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} = \alpha \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} = \alpha \mathbf{1} \quad (3.20)$$

où

$$\alpha = U^T X + c_{n-1}$$

reste à déterminer. Cependant nous avons

$$\begin{pmatrix} X \\ 1 \end{pmatrix} = \alpha F^{-1} \left(\frac{\mathbf{1}}{FC} \right),$$

la division s'effectuant composante par composante. Si nous calculons

$$Z = F^{-1} \left(\frac{\mathbf{I}}{FC} \right) = \begin{pmatrix} z_0 \\ z_1 \\ \vdots \\ z_{n-1} \end{pmatrix}$$

on obtient

$$\alpha = \frac{1}{z_{n-1}}$$

ce qui permet de calculer X. ■

La démonstration précédente, si elle a le mérite d'être constructive, ne révèle pas véritablement les propriétés des polynômes orthogonaux auxquelles elle fait implicitement appel. On peut en fait obtenir le même résultat de manière plus concise tout en conservant une approche polynômiale au moyen de l'identité de Bezout.

Supposons que la fonctionnelle cyclique c soit régulière et appliquons le théorème (1). Les polynômes P_{n+1} et P_n orthogonaux par rapport à la fonctionnelle c sont premiers entre eux et satisfont l'identité de Bezout. Donc il existe Q_{n-2} et Q_{n-1} tels que

$$P_n Q_{n-2} + P_{n-1} Q_{n-1} = c(P_{n-1}^2)$$

avec $\text{degré}(Q_{n-1}) < \text{degré}(P_n)$ et $\text{degré}(Q_{n-2}) < \text{degré}(P_{n-1})$.

Les polynômes Q_{n-2} et Q_{n-1} sont les polynômes associés respectivement à P_{n-1} et P_n . Ils sont également définis par la formule (3.2.1). Dans le cas précis où $P_n = x^n - 1$ on peut aisément calculer Q_{n-1} .

$$\begin{aligned} Q_{n-1}(x) &= c_t \left(\frac{P_n(t) - P_n(x)}{t - x} \right) \\ &= c_t \left(\frac{t^n - 1 - (x^n - 1)}{t - x} \right) \\ &= c_t \left(\frac{t^n - x^n}{t - x} \right) \\ &= c_t (x^{n-1} + x^{n-2}t + x^{n-3}t^2 + \dots + xt^{n-2} + t^{n-1}). \end{aligned}$$

Si l'on note les moments de la fonctionnelle $c(t^i) = c_i$, on obtient

$$Q_{n-1}(x) = \sum_{i=0}^{i=n-1} c_{n-1-i} x^i.$$

Remarquons que Q_{n-1} est alors le polynôme réciproque du polynôme constitué par les moments de c . L'identité de Bezout prend alors la forme suivante

$$(x^n - 1)Q_{n-2}(x) + P_{n-1}(x) \left(\sum_{i=0}^{i=n-1} c_{n-1-i} x^i \right) = c(P_{n-1}^2).$$

On remarque que l'égalité précédente permet de calculer P_{n-1} sur l'ensemble S_n des racines de l'unité d'ordre n . En supposant que Q_{n-1} ne s'annule en aucune racine de l'unité d'ordre n on peut écrire pour $i = 0, \dots, n-1$ et $w_i \in S_n$

$$P_n(w^i) = c(P_n^2)Q_{n+1}(w^i)$$

puis en reprenant les notations vectorielles des coefficients, on reconnaît en Q_{n+1} le vecteur $C^T = (c_{n-1}, c_{n-2}, \dots, c_0)$ défini précédemment. On note alors $\alpha = c(P_n^2)$ et on introduit la DFT comme opérateur d'évaluation sur les racines de l'unité d'ordre n . On retrouve ainsi la formule (3.20) dont la proposition découle.

3.6 L'algorithme régressif

3.6.1 Stratégie de l'algorithme régressif

Etant donnée une fonctionnelle c dont on connaît les $2n$ premiers moments, on commence par calculer les deux derniers polynômes orthogonaux relativement à la fonctionnelle cyclique associée \tilde{c} , celle-ci étant de taille $4n$. A partir des deux polynômes obtenus, qui sont de degré $2n$ et $2n-1$, on applique l'algorithme d'Euclide. Les restes successivement recueillis sont des polynômes orthogonaux par rapport à \tilde{c} et ceux-ci coïncident pour les degrés inférieurs ou égal à n avec les polynômes orthogonaux par rapport à c .

Algorithme 1

Les moments (c_i) , $i \in \{0, \dots, 2n-1\}$ de la fonctionnelle c sont initialement donnés. L'algorithme calcule les polynômes orthogonaux par rapport à c et dont le degré est inférieur ou égal à n .

1. Poser $P_{2n}(x) = x^{2n} - 1$.
2. Poser $C^T = (c_{2n-1}, c_{2n-2}, \dots, c_0)$.
3. Calculer $V = F^{-1}(\mathbf{I}/FC)$.
4. Poser $P_{2n-1}(x) = \sum_{i=0}^{2n-1} v_i x^i$.
5. Répéter pour $i = 2n-2, \dots, 1$,
Calculer P_i le reste de la division Euclidienne de P_{i+2} par P_{i+1}
jusqu'à ce que $P_i(x) = 0$
6. Les polynômes P_i de degré inférieur ou égal à n sont orthogonaux relativement à c .

3.6.2 Exemple dans un corps fini

L'algorithme précédent nécessite, pour sa mise en place dans un anneau de polynômes, la présence d'une transformée de Fourier discrète sur le corps de base ainsi qu'une division polynômiale avec reste. Si \mathbb{F} est un corps fini alors $\mathbb{F}[x]$ est un anneau Euclidien et donc l'algorithme d'Euclide s'applique naturellement. On dispose d'une transformée de Fourier discrète sur les corps finis ([86]). On note \mathbb{L} le corps des racines n -ièmes de l'unité sur \mathbb{F} et α une racine primitive n -ième de l'unité de \mathbb{L} , alors la transformée de Fourier discrète de dimension n est définie par l'application F_α suivante

$$\begin{aligned} \mathbb{L}[x]/(x^n - 1) &\rightarrow \mathbb{L}[X]/(X^n - 1) \\ a(x) = \sum_{i=0}^{n-1} a_i x^i &\mapsto A(x) = \sum_{j=0}^{n-1} a(\alpha^j) X^{(n-j)}. \end{aligned}$$

Cette transformation généralise aux corps finis l'évaluation polynômiale sur les racines de l'unité et peut être calculée en $O(n \log(n))$ opérations ([54], ch 4.5). F_α admet une transformation inverse lorsque n est premier avec p et

$$F_\alpha^{-1} = \frac{1}{n} F_{\alpha^{-1}}.$$

Pour ne pas faire appel à un sur-corps de \mathbb{F} , on peut sélectionner une dimension n pour laquelle une racine primitive n -ième de l'unité est dans \mathbb{F} . En considérant le sous-groupe (multiplicatif) cyclique d'ordre $p-1$ des éléments inversibles de \mathbb{F} on peut démontrer ([54], ch 4.5) que n doit diviser $p-1$ et qu'un élément générateur d'un sous-groupe cyclique d'ordre n est donné par $w^{(p-1)/n}$, où w est un générateur sous-groupe des éléments inversibles de \mathbb{F} . Construisons un exemple sur \mathbb{F}_{257} .

Exemple 9

Appliquons l'algorithme régressif pour calculer des polynômes orthogonaux par rapport à la fonctionnelle $c = (0, 0, 0, 1, 1, 1, 251, 64, \dots)$.

Une racine primitive du groupe multiplicatif de \mathbb{F}_{257} est $s = 3$. De $257 - 1 = 2^8$, on déduit qu'il existe une racine primitive 8-ième de l'unité dans \mathbb{F}_{257} et on l'obtient par

$$w = s^{\frac{p-1}{n}} \text{ mod } p = 64.$$

Cette valeur nous permet de construire la matrice de Fourier F_8 associée aux points d'interpolation $\{w, w^2, \dots, w^7, 1\}$.

$$F_8 = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 64 & 241 & 4 & 256 & 193 & 16 & 253 \\ 1 & 241 & 256 & 16 & 1 & 241 & 256 & 16 \\ 1 & 4 & 16 & 64 & 256 & 253 & 241 & 193 \\ 1 & 256 & 1 & 256 & 1 & 256 & 1 & 256 \\ 1 & 193 & 241 & 253 & 256 & 64 & 16 & 4 \\ 1 & 16 & 256 & 241 & 1 & 16 & 256 & 241 \\ 1 & 253 & 16 & 193 & 256 & 4 & 241 & 64 \end{pmatrix}.$$

Le polynôme de degré 8, orthogonal par rapport à la fonctionnelle c tronquée puis cyclitisée est

$$p_8(x) = x^8 - 1.$$

On calcule à présent le polynôme orthogonal de degré 7 par transformée de Fourier, inversion puis transformée de Fourier inverse

$$p_7(x) = 114 + 34x + 53x^2 + 3x^3 + 218x^4 + 11x^5 + 61x^6 + 235x^7.$$

L'algorithme de division Euclidienne appliqué à p_8 et p_7 nous donne

$$\begin{aligned} P_6 &= 127 + 70x + 150x^2 + 15x^3 + 169x^5 + 202x^6 \\ P_5 &= 181 + 129x + 7x^2 + 44x^3 + 212x^4 + x^5 \\ P_4 &= 145 + 127x + 92x^3 + 165x^4 \\ P_0 &= 134. \end{aligned}$$

Les polynômes orthogonaux P_1, P_2, P_3 n'existent pas pour cette fonctionnelle.

◇

3.6.3 Intérêt de l'algorithme régressif

Le principal avantage de l'algorithme régressif provient de ce que la fonctionnelle n'intervient plus au cours de la phase de progression laissant à la division Euclidienne le soin d'effectuer automatiquement les sauts dus aux éventuelles irrégularités. Les phases de transformée de Fourier discrète et d'algorithme d'Euclide peuvent être toutes les deux utilisées dans les corps finis, ce qui permet d'envisager des méthodes modulaires. Du point de vue numérique, l'algorithme régressif offre une grande variété de stratégies que nous allons développer dans le chapitre suivant.

3.6.4 Exemples numériques

Pour illustrer le comportement numérique de l'algorithme régressif nous avons comparé les calculs de la récurrence à trois termes et de l'algorithme régressif effectués en arithmétique de précision finie (16 chiffres significatifs). Les graphiques représentent le comportement de l'erreur, c'est-à-dire la différence entre les valeurs numériques en précision finie et les valeurs exactes obtenues par une programmation en calcul formel. En fonction du degré du polynôme orthogonal nous indiquons en ordonnées le logarithme à base dix de la norme Euclidienne de l'erreur

$$\log_{10} \|\tilde{P}_n - P_n\|,$$

où P_n désigne le polynôme exacte et \tilde{P}_n le polynôme obtenu en précision finie. La norme utilisée provient de \mathbb{R}^n en considérant chaque polynôme comme un vecteur dont les composantes sont les coefficients du polynôme. Lorsque l'erreur est très importante il faut

diviser par la norme du vecteur solution de manière à observer l'erreur relative qui est alors plus appropriée.

Les moments des fonctionnelles que nous avons testées (graphiques (9) et (10)) ont été générés aléatoirement, selon une loi uniforme sur l'intervalle $[0, 1]$.

Dans l'ensemble des graphiques que nous avons obtenus de cette façon nous avons discerné essentiellement deux situations différentes (graphiques (9) et (10)). Lorsque l'algorithme régressif ne rencontre pas de système intermédiaire mal conditionné dans la phase de descente due à la partie cyclique, les résultats obtenus par celui-ci sont comparables, voire meilleurs, que ceux obtenus par la récurrence à trois termes. Inversement, il se peut que l'algorithme régressif rencontre des situations intermédiaires mal conditionnées, alors la récurrence à trois termes donne de meilleurs résultats. Le cas le plus favorable à l'algorithme régressif est celui où les premiers mineurs principaux sont mal conditionnés. Si la récurrence à trois termes engendre immédiatement des erreurs qui se propagent de manière catastrophique, l'algorithme régressif, qui remonte dans la table des polynômes orthogonaux, n'est confronté à ces calculs mal conditionnés qu'en fin de parcours. C'est par exemple la situation du graphique (9). Dans le cas d'une fonctionnelle définie positive de la figure (8) la croissance du conditionnement avec le degré assure que la récurrence à trois termes est la stratégie la plus adaptée des deux. Remarquons qu'une méthode de préconditionnement est recommandée pour les matrices de Hankel définies positives car leur conditionnement devient très mauvais lorsque la dimension est grande. Un résultat de Beckermann ([5], p. 91) décrit le conditionnement minimal d'une matrice de Hankel définie positive quelconque H_{n+1} de dimension $n + 1$ par

$$\text{Cond}(H_n) \geq \frac{2}{5(n+1)^2} \exp(\delta)^{2n+2}, \quad \exp(\delta) \approx 3.210.$$

L'algorithme régressif se présente comme une stratégie complémentaire à la récurrence à trois termes. Au chapitre (4), nous allons présenter différents prolongements de cette méthode afin d'en améliorer les performances et la compréhension.

Récurrance à trois termes et algorithme régressif
pour les polynômes de Chebyshev de 2ième espèce

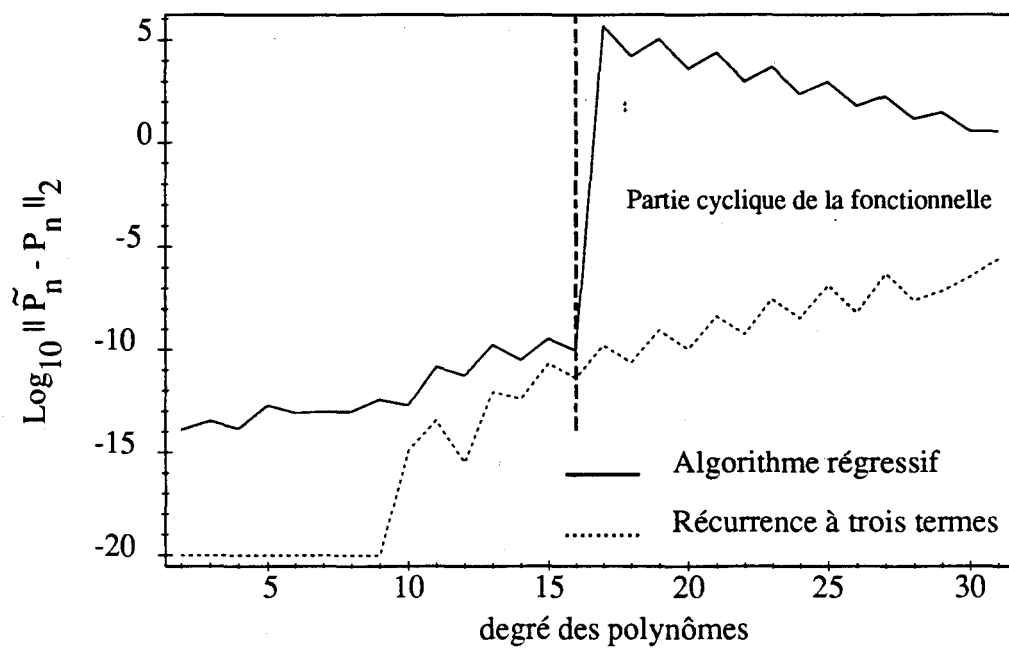


fig. 8

Récurrance à trois termes et algorithme régressif

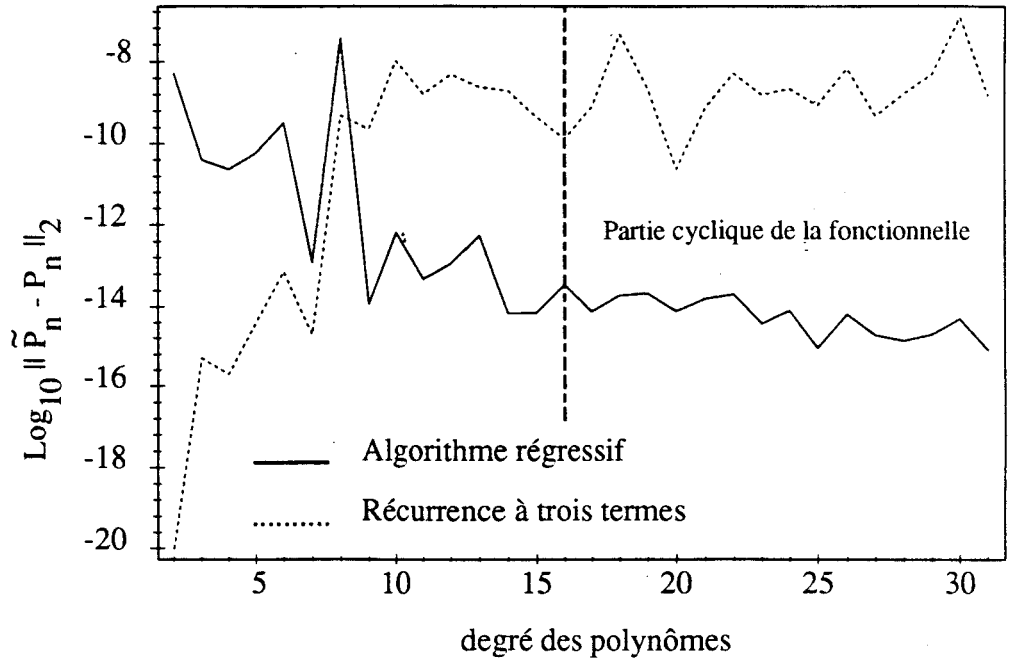


fig. 9

Récurrance à trois termes et algorithme régressif

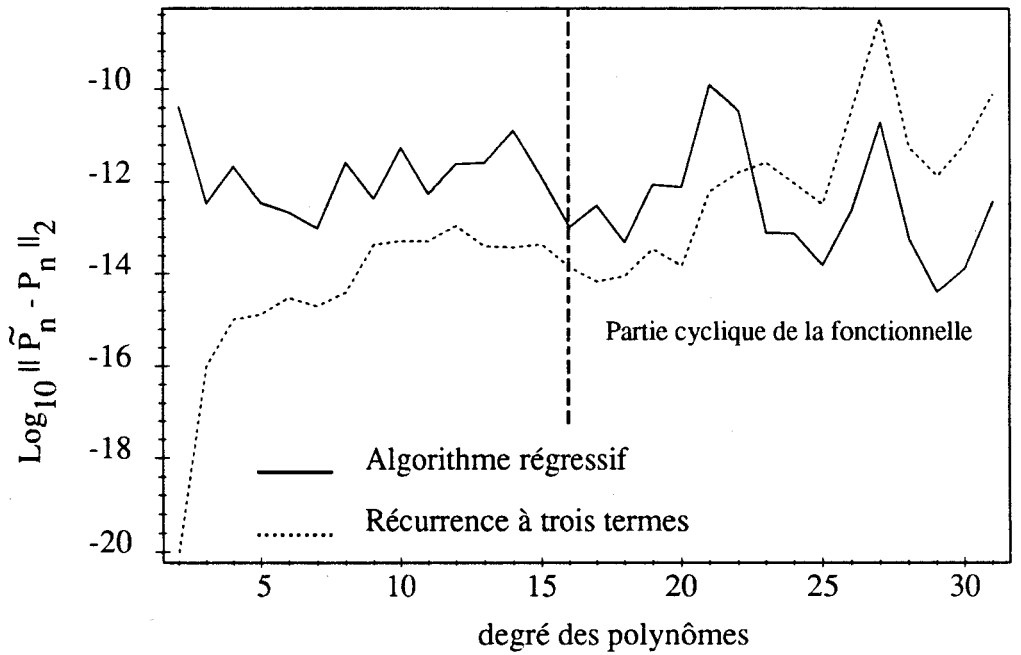


fig. 10

Chapitre 4

Extensions de l'algorithme régressif

4.1 Introduction

Dans ce chapitre nous explorons différentes modifications de l'algorithme régressif afin de contourner certaines difficultés numériques qui sont dues essentiellement au choix des représentations des polynômes. Le premier paragraphe concerne la mise en œuvre numérique de l'algorithme régressif en considérant quelques aspects de sa complexité et de sa stabilité. Une étude particulière de l'étape de division des composantes, lors du calcul de l'avant-dernier polynôme orthogonal d'une fonctionnelle cyclique, nous permet de proposer diverses méthodes pour contourner d'éventuelles divisions par zéro. L'algorithme régressif présenté par l'intermédiaire de l'interpolation sur les racines de l'unité permet d'envisager des extensions par de nouveaux choix d'ensembles de points d'interpolation. Dans cette optique, nous développons au second paragraphe deux approches de cette idée, l'une est de faire de nouveaux choix de prolongements de la fonctionnelle linéaire, l'autre se concentre sur les choix de n points d'interpolation qui maintiennent la complexité de l'interpolation sur n points en $O(n \log(n))$. Enfin, le dernier paragraphe s'intéresse à certaines représentations polynômiales et à leur aptitude à décrire la division Euclidienne.

4.2 Mise en œuvre numérique

Nous exposons brièvement les résultats connus concernant la complexité des deux étapes utilisées dans l'algorithme régressif, puis nous illustrons par un exemple l'importance de la stratégie régressive pour la stabilité du calcul numérique des polynômes orthogonaux. Entre la formulation algébrique de l'algorithme régressif et sa mise en forme numérique, il y a un certain nombre d'étapes à envisager et notamment écarter les divisions par zéro. La transformée de Fourier discrète et l'algorithme d'Euclide ont de nombreuses variantes numériques. Nous rappelons donc par des remarques les orientations possibles dans ces domaines.

$Dim(M_n)$	1	2	3	4	5	6	7	8	9	10
$Cond(M_n) \approx$	1.	5e8	5e8	2.	8.	7.	24.	34.	33.	7.

Tableau 4.1 : Conditionnements successifs.

4.2.1 Stabilité et complexité

Pour obtenir les n premiers polynômes orthogonaux relativement à une fonctionnelle c à partir de ses $2n$ premiers moments, l'algorithme régressif calcule deux FFT de vecteurs de dimension $2n$ en utilisant $O(2n \log(2n))$ opérations arithmétiques. Avec une stratégie de division pour régner, l'algorithme d'Euclide peut être réalisé en $O(2n \log(2n)^2)$ opérations. Nous insistons sur le fait que le gain effectif d'une telle stratégie par rapport à une méthode classique en $O(2n^2)$ n'est intéressant que pour de grandes dimensions et seulement pour des situations où les moments sont soit dans un corps fini, soit associés à une forme définie positive. Sur une machine parallèle les deux étapes peuvent être respectivement ramenées à $O(\log(2n))$ et $O(\log(2n)^3)$ cycles d'opérations parallèles ([72]). Le calcul des polynômes orthogonaux par la récurrence à trois termes est instable. La récurrence à trois termes suit progressivement les conditionnements des mineurs successifs donc elle peut donner une solution totalement éronnée pour un problème bien conditionné. L'algorithme régressif suit de manière régressive les mineurs principaux de la matrice de Hankel cyclique obtenue en complétant la fonctionnelle de manière cyclique. Entre P_{2n-1} , l'avant-dernier polynôme orthogonal par rapport à la fonctionnelle cyclique, et P_n , le premier polynôme orthogonal par rapport à c que l'on rencontre régressivement, tous les mineurs qui interviennent sont différents des mineurs utilisés par la récurrence à trois termes. Avec les deux algorithmes P_n est calculé comme la solution d'un même système on est donc certain que le conditionnement global de l'application qui associe P_n à c est le même dans les deux algorithmes. Cependant, les calculs intermédiaires n'ont pas le même conditionnement et le conditionnement local de chacun des chemins peut être plus mauvais que le conditionnement global. En ce cas cela signifie qu'il existe un chemin meilleur, ce qu'une méthode de pivot aurait d'ailleurs détecté. Nous allons illustrer cette situation avec le cas très important des fonctionnelles dont les petits mineurs sont mal conditionnés.

Exemple 10

Pour la fonctionnelle c dont les premiers moments sont

$$\{1., 10^{-9}, 2.10^{-9}, -10^{-8}, 1., -0.4, -0.02, 0.56, -0.18, 0.8, \dots\},$$

on constate que les premiers mineurs sont mal conditionnés alors que les mineurs qui interviennent dans l'algorithme régressif pour le calcul de P_4 et P_5 sont bien mieux conditionnés. Les conditionnements des mineurs successifs de la matrice de Hankel cyclique obtenue à partir de c sont donnés dans le tableau suivant. L'algorithme régressif est donc ici à privilégier. Cette circonstance se rencontre tout particulièrement quand on souhaite utiliser l'algorithme récursif du chapitre (5) pour faire un saut au-dessus des étapes mal

n	Récurrance	Régressif
1	0	.1e-13
2	0	.5e-4
3	0	4333.7
4	.9e-6	.1e-13
5	.2e-5	.1e-12

Tableau 4.2 : Comparaison récurrence et régressif.

conditionnées. Numériquement les erreurs suivent en général le plus mauvais conditionnement rencontré par l'algorithme. Nous avons calculé les valeurs des polynômes orthogonaux par rapport à c en arithmétique exacte et nous comparons dans le tableau (4.2), les erreurs e_n commises sur les polynômes orthogonaux obtenus numériquement dans les deux situations avec une précision machine de 16 chiffres significatifs. L'erreur e_n sur le polynôme de degré n est définie par

$$e_n = \sqrt{\sum_{i \leq n} |p_i^{(n)} - v_i^{(n)}|^2}, \quad (4.1)$$

où les coefficients $p_i^{(n)}$ sont ceux du polynôme de degré n obtenu en arithmétique de précision finie et les coefficients $v_i^{(n)}$ sont ceux du polynôme de degré n obtenu en arithmétique exacte.

◇

4.2.2 La DFT et les divisions

Dans l'algorithme régressif, l'étape de division n'est pas toujours réalisable. C'est le cas lorsque FC a des composantes qui s'annulent. Pour que cela n'arrive pas, il faut que la matrice de convolution associée à C soit inversible, ou encore que C considéré comme polynôme n'admette pas pour racine une des racines de l'unité d'ordre $2n - 1$. Remarquons que l'existence de P_{2n} et P_{2n-1} pour la fonctionnelle cyclique n'assure pas celle de P_n comme dans l'exemple suivant.

Exemple 11

Pour la fonctionnelle cyclique $c = \{1, 2, 3, 4, 5, 2, 1, 2, 3, 4, 5, 2\}$, on a

$$\begin{vmatrix} 1 & 2 & 3 \\ 2 & 3 & 4 \\ 3 & 4 & 5 \end{vmatrix} = 0$$

et c n'a pas de polynôme orthogonal régulier de degré 3. Pourtant la fonctionnelle cyclique

admet des polynômes orthogonaux de degré 5 et 6 donnés par

$$P_5(t) = -\frac{3t^2}{31} + \frac{8t}{93} + t^5 - \frac{43t^4}{93} + \frac{25t^3}{93} - \frac{20}{31}$$

et

$$P_6(t) = t^6 - 1.$$

◇

4.2.3 Comment éviter les divisions par zéro

Pour simplifier les notations, nous considérons que c est une fonctionnelle cyclique donnée par les moments $(c_0, \dots, c_n, c_0, \dots, c_n)$. La première division est due à l'étape de déconvolution. Si l'un des dénominateurs est nul, cela engendre un blocage de l'algorithme, P_{n-1} ne pouvant d'ailleurs plus être calculé. On note respectivement F_n et F_{n+1} les matrices de Fourier d'ordre n et $n+1$, S_n et S_{n+1} les ensembles de racines de l'unité dans \mathbb{C} d'ordre n et $n+1$. Nous reprenons l'identité de Bezout (3.1) sur l'ensemble des racines n -ièmes de l'unité. Sachant que $P_n = x^n - 1$, cela nous conduit à

$$P_{n-1}(w)Q_{(n-1)}(w) = 1 \text{ pour tout } w \in S_n,$$

avec $Q_{n-1} = c_{n-1} + c_{n-2}x + \dots + c_1x^{n-2} + c_0x^{n-1}$. Si une composante de $F_n C$ s'annule c'est que $Q_{(n-1)}(w) = 0$ pour un certain $w \in S_n$, alors $P_{n-1}(w)$ ne peut pas exister et on ne peut pas appliquer l'algorithme régressif.

Supposons que l'une des composantes de $F_n C$ s'annule. Dans ce cas on ajoute un moment c_{n+1} à la fonctionnelle c de moments $(c_i), i = 0, \dots, n$, de manière à obtenir une fonctionnelle d qui donne les mêmes polynômes orthogonaux. En appliquant l'algorithme à la fonctionnelle cyclique associée à d on est amené à calculer $F_{n+1}D$ où $D^T = (c_{n+1}, c_n, \dots, c_0)$. Cependant, $F_{n+1}D$ peut cette fois être choisi sans composante nulle grâce au paramètre c_{n+1} . Plus précisément, si on considère les polynômes

$$Q_n(x) = \sum_{i=0}^{i=n} c_{n-i}x^i \text{ et } Q_{n+1}(x) = \sum_{i=0}^{i=n+1} c_{n+1-i}x^i$$

nous avons alors la relation

$$c_{n+1} + xQ_n(x) = Q_{n+1}(x).$$

Les composantes nulles de $F_n C$ et de $F_{n+1}D$ s'interprètent comme des zéros de $Q_n(x)$ et de $Q_{n+1}(x)$ localisés sur S_n et S_{n+1} respectivement. Pour préserver $F_{n+1}D$ des composantes nulles, il suffit d'évaluer le polynôme $xQ_n(x)$ sur les racines de l'unité d'ordre $n+1$ et d'en déduire un choix approprié de c_{n+1} . On pourrait prendre par exemple

$$c_{n+1} = \max_{\{z \in S_{n+1}\}} |zQ_n(z)| + 1.$$

On peut aussi recourir au choix de c_{n+1} pour mieux répartir les composantes de $F_{n+1}D$, en évitant par exemple des valeurs trop proches de zéro et des valeurs trop grandes. Dans cette optique, si on note

$$\delta_s = \min_{\substack{z \in S_{n+1} \\ zQ_n(z) \neq sQ_n(s)}} |zQ_n(z) - sQ_n(s)|,$$

on peut déterminer la racine de l'unité s_0 qui donne la valeur de $xQ_n(x)$ la plus éloignée des autres, à savoir le nombre s_0 tel que

$$\delta_{s_0} = \max_{z \in S_{n+1}} \delta_z.$$

Pour atteindre notre objectif on peut opter pour

$$c_{n+1} = -\frac{1}{2}(z_0Q_n(z_0) + s_0Q_n(s_0)),$$

où z_0 réalise la minimisation

$$|z_0Q_n(z_0) - s_0Q_n(s_0)| = \delta_{s_0}.$$

Ces précautions garantissent que pour tout $z \in S_{n+1}$

$$\begin{aligned} |Q_{n+1}(z)| &= |c_{n+1} + zQ_n(z)| \\ &= |zQ_n(z) - \frac{1}{2}(z_0Q_n(z_0) + s_0Q_n(s_0))| \\ &= |\frac{1}{2}(zQ_n(z) - s_0Q_n(s_0)) + \frac{1}{2}(zQ_n(z) - z_0Q_n(z_0))| \\ &\geq \frac{1}{2}|z_0Q_n(z_0) - s_0Q_n(s_0)| \\ &\geq \frac{1}{2}\delta_{s_0}, \end{aligned}$$

et ce en vertu de l'inégalité triangulaire. Malheureusement ces calculs sont trop coûteux. Des considérations géométriques plus accessibles peuvent donner de bons résultats. On peut privilégier l'option de maintenir l'ensemble des valeurs $\{zQ_n(z), z \in S_{n+1}\}$ aussi rapproché que possible de zéro, quitte à faire une petite translation après coup pour éviter les valeurs trop proches de zéro. Ce qui importe le plus c'est en effet d'éloigner les racines de P_{n-1} des racines de l'unité, ou d'éloigner de zéro les valeurs de $P_{n-1}(w)$, $w \in S_n$.

Les deux stratégies que nous venons de proposer sont complémentaires. En effet, si les valeurs de $zQ_n(z)$ sur S_{n+1} sont fortement regroupées autour de zéro alors δ_{s_0} est petit et n'offre pas la sécurité numérique recherchée. Par contre, la première méthode éloigne de zéro l'ensemble des valeurs de $zQ_n(z)$ précédentes. Les translations que nous venons de proposer sur l'ensemble des valeurs $\{zQ_n(z), z \in S_{n+1}\}$ ne permettent pas d'améliorer les situations où il y a simultanément des valeurs éloignées de zéro et des valeurs proches de

zéro. Il faut compléter la méthode par une homothétie sur les valeurs $\{zQ_n(z), z \in S_{n+1}\}$. Puisque les coefficients de $Q(z)$ dépendent linéairement de c , en multipliant les moments de la fonctionnelle c par un coefficient λ on multiplie d'autant les valeurs de $zQ_n(z)$. On contrôle alors la dispersion des valeurs de $zQ_n(z)$ par l'homothétie de centre $z_0 = o$ et de rapport λ et on maintient les valeurs à distance de zéro par translation de c_{n+1} , tout cela sans modifier théoriquement les polynômes orthogonaux cherchés.

Après avoir choisi une translation qui recentre les valeurs $Q_{n+1}(z)$, $z \in S_{n+1}$ autour de zéro, pour optimiser le coefficient d'homothétie λ on peut prendre

$$\lambda \max_{z \in S_{n+1}} Q_{n+1}(z) = \frac{1}{\lambda \min_{z \in S_{n+1}} Q_{n+1}(z)}.$$

De cette façon, l'inversion utilisée pour le calcul des $P_{n+1}(z)$, $z \in S_{n+1}$, maintient les valeurs dans la couronne qui contient les $Q_{n+1}(z)$, $z \in S_{n+1}$.

On peut également envisager le contraire, en ramenant par une homothétie toute les valeurs $zQ_n(z)$, $z \in S_{n+1}$ dans le cercle centré en zéro et de rayon $1/2$. Puis une translation imposée par le choix $c_{n+1} = 1$ maintient les valeurs $Q_{n+1}(z)$, $z \in S_{n+1}$ dans une couronne centrée en zéro, de petit rayon $1/2$ et de grand rayon $3/2$. Les valeurs $P_{n+1}(z)$, $z \in S_{n+1}$ sont alors contenues dans la couronne centrée en zéro, de petit rayon $2/3$ et de grand rayon 2 . Nous illustrons par un exemple l'intérêt d'une meilleure répartition des valeurs de $zQ_n(z)$ sur S_{n+1} .

Exemple 12

Prenons le cas de la fonctionnelle c , donnée par les moments

$$\{-33/100, -1/25, 41/100, -1101/2500, 3/5, -2/5, -1/50, 14/25, -9/50, 4/5, \dots\},$$

et qui, une fois rendue cyclique, admet un polynôme $Q_9(t)$, associé de $F_{10}(t) = t^{10} - 1$. Malheureusement, $Q_9(t)$ prend une valeur de l'ordre de $-0.4e - 3$ en $t = -1$ une racine de l'unité de S_{10} . Inverser cette valeur pour reconstituer P_9 , le polynôme de degré 9 orthogonal par rapport à la fonctionnelle cyclique, génère une perte d'information numérique. Pour pallier à cela, nous proposons de recentrer les valeurs de $tQ_9(t)$ sur S_{11} autour de l'origine, puis par une homothétie sur les coefficients de c , nous les ramenons dans un cercle de rayon $1/2$. Sur les figures (11) et (12) nous représentons respectivement la répartition des valeurs de $Q_9(t)$ sur S_{10} et, après une homothétie et l'ajout d'un moment c_{10} à la fonctionnelle, la répartition des valeurs de $Q_{10}(t)$ sur S_{11} . Voyons comment nous avons choisi c_{10} . Pour l'exemple en question nous avons calculé le centre de gravité de l'ensemble des valeurs $tQ_9(t)$ sur S_{11} en leur affectant des masses égales aux modules de chaque valeur. En prenant ce point pour origine, nous devrions en général améliorer la répartition des valeurs autour de l'origine. Lorsque les moments sont réels, le centre de gravité est réel, la symétrie des parties imaginaires étant due à la conjugaison des racines de l'unité. Dans notre exemple, ce centre est environ $0.4e - 1$ ce qui signifie que les valeurs $tQ_9(t)$ sur S_{11} sont déjà bien réparties autour de l'origine et que la translation avant l'homothétie n'apporte rien. L'homothétie de rapport environ $k \approx 0.213$ centrée en l'origine ramène

n	Régressif	Modifié
1	.490e-11	.497e-13
2	.138e-10	.183e-12
3	.898e-11	.195e-12
4	.418e-10	.423e-12
5	.417e-10	.298e-12

Tableau 4.3 : Comparaison régressif et modifié.

toutes les valeurs dans le disque de rayon $1/2$ centré en l'origine. Une translation de 1 déplace toutes les valeurs dans un disque de rayon $1/2$ centré en 1 qui ne contient pas l'origine. L'algorithme régressif appliqué dans chacun des cas montre clairement que l'on a tout intérêt à maintenir les valeurs de $Q_9(t)$ sur S_{10} à l'écart de zéro, disons à une distance supérieure à $10^{(-1)}$, et en même temps pas trop loin, par exemple à une distance inférieure à 2 . En reprenant la formule d'erreur utilisée en (4.1) nous obtenons le tableau comparatif (4.3).

Nous proposons un autre exemple qui justifie l'utilisation de la similitude précédente dans le cas d'arithmétique exacte et nous comparons de nouveau les résultats numériques. Nous avons artificiellement créé une difficulté supplémentaire en partant d'une situation où $Q_9(t)$ admet une racine dans S_{10} et une racine dans S_{11} . Cette fois-ci l'algorithme régressif ne peut pas s'appliquer sans l'utilisation d'une des méthodes précédentes.

Exemple 13

La fonctionnelle $c = \{-2, 1, 1, 0, 2, -2, 1, 2, -2, -1\}$ a été construite de telle sorte que $t^2 - 1$ divise $Q_9(t)$ et possède donc -1 comme racine dans S_{10} . 1 est alors racine de $tQ_9(t)$ dans S_{11} . Non seulement l'algorithme régressif ne peut pas s'appliquer mais en plus on est obligé d'ajouter un moment non nul si on veut appliquer l'algorithme régressif sur une dimension plus grande. En appliquant la méthode précédente, on a pour centre de gravité $g \approx 3$ et un rapport d'homothétie $k \approx 0.6e - 1$. Les erreurs (4.1) de la version numérique sont données dans le tableau suivant (4.4).

n	Régressif	Modifié
1	-	.882e-14
2	-	.193e-13
3	-	.318e-13
4	-	.632e-13
5	division par 0	.743e-13

Tableau 4.4 : Intérêt de la modification.

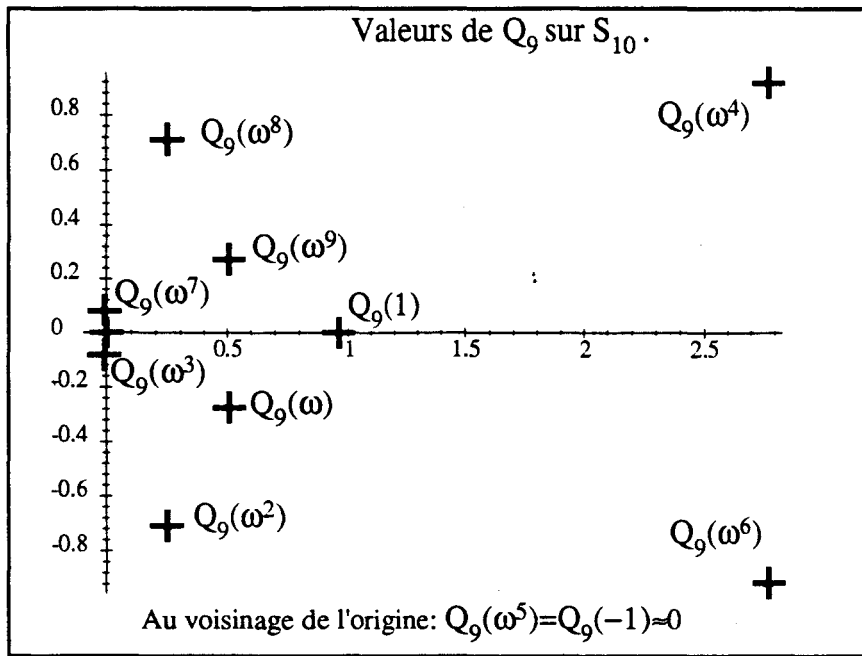


fig. 11

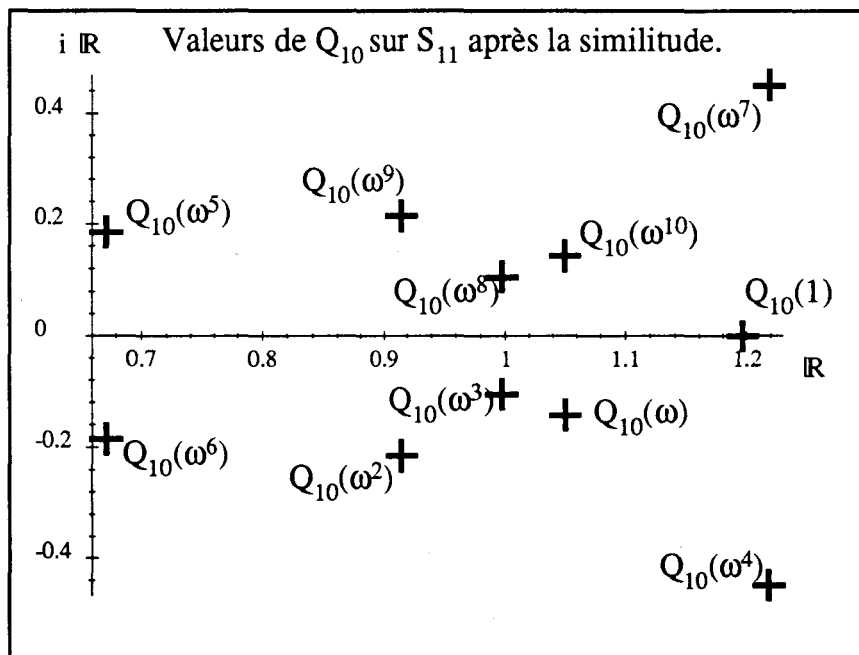


fig. 12

4.3 Autres procédés régressifs

Nous explorons à présent deux méthodes différentes pour généraliser l'algorithme régressif. A partir du principe de prolongement cyclique on a cherché des prolongements variés en s'intéressant à l'interpolation ou à l'interprétation en terme de bordage.

4.3.1 Autres prolongements périodiques de la fonctionnelle

L'idée de départ utilisée dans la méthode régressive est de compléter la fonctionnelle de manière à pouvoir calculer les derniers polynômes orthogonaux pour la fonctionnelle obtenue puis de redescendre par division Euclidienne. Une première modification consiste à compléter différemment la fonctionnelle. En s'inspirant de la méthode de calcul du dernier polynôme, on étend le choix du second membre du système (1.4) à celui de n'importe quelle colonne de la matrice de Hankel du système. On construit alors la fonctionnelle qui satisfait cette contrainte. On doit donc choisir le vecteur $(a_0, a_1, \dots, a_{2n-1})^T$ de manière à ce que le système de Hankel suivant

$$\begin{pmatrix} c_0 & c_1 & \cdots & c_{2n-2} & c_{2n-1} \\ c_1 & c_2 & \cdots & c_{2n-1} & a_0 \\ \vdots & \vdots & & \vdots & \vdots \\ c_{2n-2} & c_{2n-1} & \cdots & a_{2n-4} & a_{2n-3} \\ c_{2n-1} & a_0 & \cdots & a_{2n-3} & a_{2n-2} \end{pmatrix} \begin{pmatrix} p_0 \\ p_1 \\ \vdots \\ \vdots \\ p_{2n-1} \end{pmatrix} = - \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ \vdots \\ a_{2n-1} \end{pmatrix} \quad (4.2)$$

ait pour second membre par exemple la $k + 1$ -ième colonne de la matrice de Hankel. On doit alors avoir l'égalité

$$\begin{pmatrix} c_k \\ c_{k+1} \\ \vdots \\ c_{2n-1} \\ a_0 \\ \vdots \\ a_{k-1} \end{pmatrix} = - \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_{2n-1-k} \\ a_{2n-k} \\ \vdots \\ a_{2n-1} \end{pmatrix}. \quad (4.3)$$

Les $2n - k$ premières équations déterminent de façon unique les moments $a_0, a_1, \dots, a_{2n-1-k}$, puis en reportant ces valeurs dans le premier membre de l'égalité (4.3), on obtient les $2n - k$ moments suivants. En procédant par récurrence on obtient la totalité des moments et ce de manière unique. Le procédé de prolongement obtenu consiste ainsi à reporter périodiquement les $2n - k$ derniers moments de la fonctionnelle de départ pour doubler le nombre de moments. Avec cette configuration, la solution du système (4.2) est par construction

$$\begin{cases} p_i = 0 & \text{pour } i \neq k \\ p_k = 1 & \end{cases}$$

Le polynôme orthogonal de degré $2n$ pour cette fonctionnelle est alors

$$P_{2n}(x) = x^{2n} - x^k.$$

Remarquons alors qu'il est possible de recueillir un prolongement des moments qui donne pour dernier polynôme orthogonal pour cette fonctionnelle un polynôme fixé quelconque.

4.3.2 Prolongements quelconques de la fonctionnelle

Si l'on désire obtenir $P_{2n}(x) = x^{2n} + \sum_{i=0}^{2n-1} p_i x^i$ comme dernier polynôme orthogonal, le prolongement à effectuer pour la fonctionnelle est une solution $(a_0, a_1, \dots, a_{2n-1})$ du système (4.2) que l'on peut encore écrire

$$M = \begin{pmatrix} c_0 & c_1 & \cdots & c_{2n-2} & c_{2n-1} \\ c_1 & c_2 & \cdots & c_{2n-1} & 0 \\ \vdots & \vdots & & 0 & \vdots \\ c_{2n-2} & c_{2n-1} & 0 & \vdots & 0 \\ c_{2n-1} & 0 & \cdots & 0 & 0 \end{pmatrix} \begin{pmatrix} p_0 \\ p_1 \\ \vdots \\ \vdots \\ p_{2n-1} \end{pmatrix} \quad (4.4)$$

$$= - \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ \vdots \\ a_{2n-1} \end{pmatrix} - \begin{pmatrix} 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & a_0 \\ \vdots & \vdots & & a_0 & \vdots \\ 0 & 0 & a_0 & \vdots & a_{2n-3} \\ 0 & a_0 & \cdots & a_{2n-3} & a_{2n-2} \end{pmatrix} \begin{pmatrix} p_0 \\ p_1 \\ \vdots \\ \vdots \\ p_{2n-1} \end{pmatrix}. \quad (4.5)$$

L'écriture du produit d'une matrice de Toeplitz triangulaire inférieure par un vecteur, de par sa similitude avec le produit de polynômes (3.12), prend la forme suivante

$$M = \begin{pmatrix} c_0 & c_1 & \cdots & c_{2n-2} & c_{2n-1} \\ c_1 & c_2 & \cdots & c_{2n-1} & 0 \\ \vdots & \vdots & & 0 & \vdots \\ c_{2n-2} & c_{2n-1} & 0 & \vdots & 0 \\ c_{2n-1} & 0 & \cdots & 0 & 0 \end{pmatrix} \begin{pmatrix} p_0 \\ p_1 \\ \vdots \\ \vdots \\ p_{2n-1} \end{pmatrix} \quad (4.6)$$

$$= - \begin{pmatrix} 1 & 0 & \cdots & 0 & 0 \\ p_{2n-1} & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & & \vdots & \vdots \\ p_2 & p_3 & \cdots & 1 & 0 \\ p_1 & p_2 & \cdots & p_{2n-1} & 1 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ \vdots \\ a_{2n-1} \end{pmatrix}. \quad (4.7)$$

On peut ainsi observer que la donnée de P_{2n} et des coefficients de la fonctionnelle permet de calculer les moments restants. On a donc le moyen d'obtenir pour des prolongements

variés le dernier polynôme orthogonal pour la fonctionnelle. De la même manière, si on se donne la fin de la fonctionnelle et le dernier polynôme orthogonal pour celle-ci alors on peut calculer le début de la fonctionnelle. Il suffit pour le voir d'écrire la formule (4.6) sous la forme

$$M = \begin{pmatrix} p_0 & p_1 & \cdots & p_{2n-2} & p_{2n-1} \\ 0 & p_0 & \cdots & & p_{2n-2} \\ \vdots & 0 & & & \vdots \\ 0 & \vdots & 0 & p_0 & p_1 \\ 0 & 0 & \cdots & 0 & p_0 \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ \vdots \\ \vdots \\ c_{2n-1} \end{pmatrix}. \quad (4.8)$$

Voyons maintenant comment obtenir le polynôme orthogonal précédent.

L'identité de Bezout et la réciprocité

Rappelons que les polynômes unitaires P_{2n} et P_{2n-1} vérifient l'identité de Bezout,

$$P_{2n}Q_{2n-2} - P_{2n-1}Q_{2n-1} = c(P_{2n-1}^2)$$

où Q_{2n-2} et Q_{2n-1} sont les polynômes associés de P_{2n} et P_{2n-1} par la fonctionnelle c . Le calcul d'un polynôme associé $Q_{2n-1}(x) = \sum_{i=0}^{2n-1} q_i x^i$ ne nécessite que les premiers moments de la fonctionnelle, et l'opération est détaillée en (3.3). On a

$$\begin{pmatrix} c_0 & c_1 & \cdots & c_{2n-2} & c_{2n-1} \\ 0 & c_0 & \cdots & c_{2n-3} & c_{2n-2} \\ \vdots & 0 & & \vdots & \vdots \\ 0 & \vdots & & c_0 & c_1 \\ 0 & 0 & \cdots & 0 & c_0 \end{pmatrix} \begin{pmatrix} p_1 \\ p_2 \\ \vdots \\ \vdots \\ p_{2n} \end{pmatrix} = \begin{pmatrix} q_0 \\ q_1 \\ \vdots \\ \vdots \\ q_{2n-1} \end{pmatrix}. \quad (4.9)$$

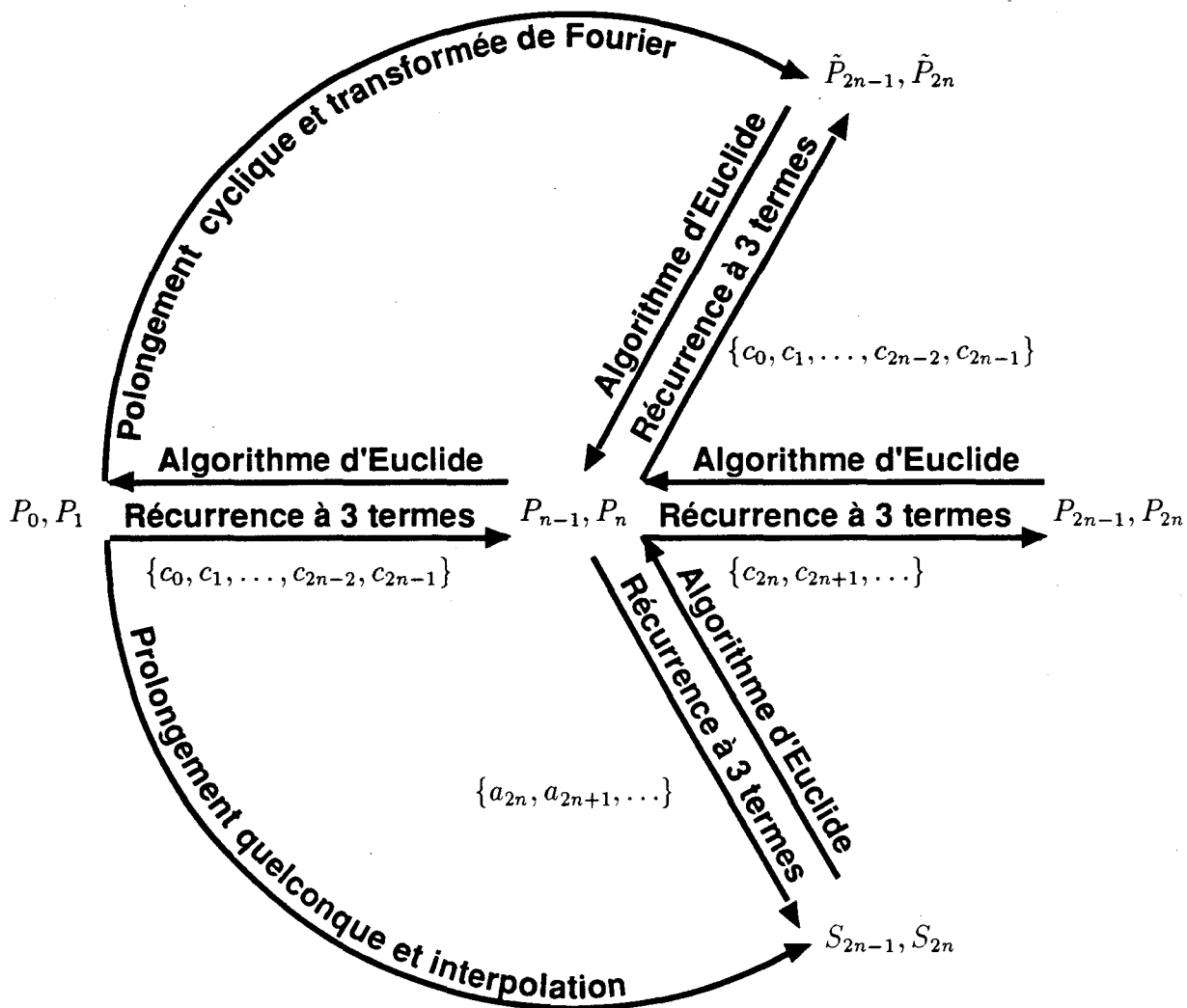
Connaissant P_{2n} on obtient donc Q_{2n-1} . La recherche de P_{2n-1} peut se formuler alors comme un problème de réciprocité puisque P_{2n-1} est le polynôme réciproque du polynôme associé $-Q_{2n-1}$

$$-P_{2n-1}Q_{2n-1} \equiv 1 \pmod{P_{2n}}. \quad (4.10)$$

L'interpolation et l'évaluation polynomiale

A partir de la situation précédente (4.10), on évalue Q_{2n-1} sur les racines de P_{2n} , puis on calcule par inversion les valeurs de P_{2n-1} sur les racines de P_{2n} . Par interpolation sur les racines de P_{2n} on récupère alors P_{2n-1} (13). Néanmoins, l'interpolation n'est pas une opération très stable si les noeuds d'interpolation sont mal répartis. Dans le cas de racines multiples pour P_{2n} on doit faire appel aux formules d'interpolation d'Hermite ([98]). La méthode de Lagrange peut être avantageusement employée à l'aide des fractions partielles ([43]). On dispose également de l'algorithme d'Euclide étendu qui, appliqué à P_{2n} et Q_{2n-1} , donnera P_{2n-1} . On aura intérêt à choisir P_{2n} de manière à rendre cette étape la plus stable possible.

fig. 13: Algorithmes régressifs



Dans les situations qui font usage de la DFT on considère que l'on s'est ramené au préalable à $n = 2^k$ en augmentant le nombre de moments. Dans le cas de l'algorithme régressif, le choix de ces moments doit satisfaire au mieux la contrainte suivante. La FFT de la fonctionnelle obtenue ne doit pas s'annuler et de préférence se rapprocher de $\mathbb{1}$ par valeurs supérieures. De cette façon P_{2n-1} n'a pas de racine commune avec $P_{2n} = x^{2n} - 1$ et on peut espérer avoir mis une distance correcte entre les racines des deux polynômes. Nous étudions à présent quelques cas simples.

4.3.3 Choix particuliers du dernier polynôme orthogonal

1. $P_{2n}(x) = x^{2n} - 1$. Ce choix correspond bien sûr au cas (3.18) obtenu en prolongeant la fonctionnelle de manière cyclique. La matrice des moments est de type circulante à l'ordre $2n$. En résumé, P_{2n-1} s'obtient par la FFT en $O(n \log(n))$.
2. $P_{2n}(x) = x^{2n}$. Lorsqu'on impose cette condition, le vecteur composé par les coefficients de P_{2n} , (p_0, \dots, p_{2n-1}) est le vecteur nul. La matrice des moments à l'ordre $2n$ est de type Toeplitz triangulaire. Le calcul du polynôme associé Q_{2n-1} présente un coût en $O(n \log(n))$. Déterminons le polynôme P_{2n-1} .

L'équation de réciprocity devient

$$-P_{2n-1}Q_{2n-1} = 1 \pmod{x^{2n}}.$$

Ce problème est équivalent à la division polynomiale ou encore à l'inversion d'une matrice de Toeplitz triangulaire, il ne nécessite donc que $O(n \log(n))$ opérations.

3. $P_{2n}(x) = x^{2n} - \alpha$ avec $\alpha \neq 0$. On obtient ce polynôme pour une matrice circulante altérée par le coefficient α . La matrice des moments est

$$\begin{pmatrix} c_0 & c_1 & \cdots & c_{2n-2} & c_{2n-1} \\ c_1 & c_2 & \cdots & c_{2n-1} & \alpha c_0 \\ \vdots & \vdots & & \vdots & \vdots \\ c_{2n-2} & c_{2n-1} & \cdots & & \alpha c_{2n-3} \\ c_{2n-1} & \alpha c_0 & \cdots & \alpha c_{2n-3} & \alpha c_{2n-2} \end{pmatrix}. \quad (4.11)$$

Le polynôme P_{2n-1} cherché doit vérifier

$$(x^{2n} - \alpha)Q_{2n-2}(x) - P_{2n-1}(x)Q_{2n-1}(x) = 1. \quad (4.12)$$

Soit β un nombre complexe tel que $\beta^{2n} = \alpha$, on peut considérer P_{2n} comme un polynôme de la variable x/β .

$$P_{2n}(x) = x^{2n} - \alpha = \alpha \left(\frac{x^{2n}}{\alpha} - 1 \right) = \alpha \left(\left(\frac{x}{\beta} \right)^{2n} - 1 \right)$$

En changeant également la variable dans les autres polynômes de l'égalité (4.12), on obtient

$$\left(\left(\frac{x}{\beta} \right)^{2n} - 1 \right) \tilde{Q}_{2n-2} \left(\frac{x}{\beta} \right) - \tilde{P}_{2n-1} \left(\frac{x}{\beta} \right) \tilde{Q}_{2n-1} \left(\frac{x}{\beta} \right) = \frac{1}{\alpha}.$$

On se retrouve dans la situation (1) où la FFT permet de trouver $\tilde{P}_{2n-1}(x/\beta)$ avec une complexité en $O(n \log_2(n))$.

4. $P_{2n}(x) = x^{2n} - x^k$ avec $k < 2n$. Ce choix provient d'un prolongement périodique de la fonctionnelle dont la forme est présentée en (4.3). La famille des moments est de la forme suivante,

$$c_0, c_1, \dots, c_k, \dots, c_{2n-2}, c_k, \dots, c_{2n-2}, \dots.$$

P_{2n-1} est alors solution de

$$-P_{2n-1}Q_{2n-1} = 1 \pmod{x^{2n} - x^k}.$$

En remarquant la factorisation $x^{2n} - x^k = (x^{2n-k} - 1)x^k$, l'équation apparaît comme une combinaison des choix (1) et (2). Les contraintes d'interpolation sont réparties sur l'origine et les racines de l'unité d'ordre k . On peut notamment décomposer la difficulté à l'aide du système suivant

$$\begin{cases} -P_{2n-1}Q_{2n-1} = 1 \pmod{x^k} \\ -P_{2n-1}Q_{2n-1} = 1 \pmod{x^{2n-k} - 1}. \end{cases} \quad (4.13)$$

Ecrivons P_{2n-1} et Q_{2n-1} sous la forme

$$P_{2n-1}(x) = U(x) + x^k V(x)$$

$$Q_{2n-1} = S(x) + x^k T(x)$$

avec des polynômes U, S de degré $k - 1$ puis V et T de degré $2n - k - 1$. Le polynôme associé Q_{2n-1} étant calculé, S et T sont maintenant fixés et le produit $P_{2n-1}Q_{2n-1}$ s'exprime en fonction des polynômes inconnus U et V

$$P_{2n-1}(x)Q_{2n-1}(x) = U(x)S(x) + x^k(U(x)T(x) + V(x)S(x)) + x^{2k}V(x)T(x).$$

On en déduit

$$U(x)S(x) = -1 \pmod{x^k}.$$

On évalue ensuite U conformément au choix (2), ce qui produit un coût d'ordre $O(k \log_2(k))$. La deuxième équation de (4.13) devient

$$V(x)(x^k S(x) + x^{2k} T(x)) = 1 - U(x)S(x) - x^k U(x)T(x) \pmod{x^{2n-k} - 1}.$$

On obtient V sur base des résultats obtenus au choix 1. Le coût de cette dernière opération est de l'ordre de $O((2n - k) \log_2(2n - k))$ en employant la FFT. Avec ce choix, on peut finalement se procurer les deux derniers polynômes orthogonaux pour un coût total en $O(n \log(n))$.

5. $P_{2n}(x) = x^{2n} - \alpha x^k$ avec $k < 2n$. Ceci correspond à un prolongement périodique de type (4) perturbé par le paramètre α . La famille des moments a , pour la circonstance, l'aspect suivant

$$c_0, c_1, \dots, c_k, \dots, c_{2n-2}, \alpha c_k, \dots, \alpha c_{2n-2}, \alpha c_k, \dots.$$

P_{2n-1} est alors solution de

$$-P_{2n-1}Q_{2n-1} = 1 \pmod{x^{2n} - \alpha x^k}.$$

En se référant au cas précédent, la factorisation $x^{2n} - \alpha x^k = (x^{2n-k} - \alpha)x^k$, l'équation apparaît comme une combinaison des choix 2 et 3. La décomposition qui s'impose est la suivante

$$\begin{cases} -P_{2n-1}Q_{2n-1} = 1 \pmod{x^k} \\ -P_{2n-1}Q_{2n-1} = 1 \pmod{x^{2n-k} - \alpha}. \end{cases} \quad (4.14)$$

En reprenant la même écriture de P_{2n-1} et Q_{2n-1} , on obtient à nouveau U à l'aide de

$$U(x)S(x) = -1 \pmod{x^k}.$$

En ce qui concerne V , on a de manière similaire

$$V(x)(x^k S(x) + x^{2k} T(x)) = 1 - U(x)S(x) - x^k U(x)T(x) \pmod{x^{2n-k} - \alpha}.$$

En s'inspirant des résultats obtenus au choix 3, le coût de ce calcul est encore de l'ordre de $O((2n-k) \log_2(2n-k))$ en employant la FFT. Le coût total en $O(n \log(n))$ est maintenu.

6. $P_{2n}(x) = T_{2n}(x)$ où T_{2n} désigne le polynôme de Tchebyshev de degré $2n$. Ce cas est tout à fait semblable au choix (1), mais s'adapte mieux aux polynômes de $\mathbb{R}[x]$ et sa complexité reste en $O(n \log(n))$.

4.3.4 Choix particuliers de l'avant-dernier polynôme orthogonal

Maintenant, on se donne le polynôme unitaire P_{2n-1} et on essaye de reconstituer P_{2n} . Le système qui détermine la fonctionnelle prolongée à partir de P_{2n-1} est

$$\begin{pmatrix} c_0 & c_1 & \cdots & c_{2n-3} & c_{2n-2} \\ c_1 & c_2 & \cdots & c_{2n-2} & c_{2n-1} \\ \vdots & \vdots & & \vdots & a_0 \\ c_{2n-3} & c_{2n-2} & \cdots & & \vdots \\ c_{2n-2} & c_{2n-1} & a_0 & \cdots & a_{2n-4} \end{pmatrix} \begin{pmatrix} p_0 \\ p_1 \\ \vdots \\ \vdots \\ p_{2n-2} \end{pmatrix} = - \begin{pmatrix} c_{2n-1} \\ a_0 \\ \vdots \\ \vdots \\ a_{2n-3} \end{pmatrix}. \quad (4.15)$$

En observant le système on se rend compte que l'on a moins de liberté pour le choix de P_{2n-1} . Le lien établi entre c_{2n-1} et c_0, \dots, c_{2n-2} par la première ligne du système impose naturellement une contrainte sur les coefficients de P_{2n-1} . On est ainsi conduit à glisser un paramètre dans les types de polynômes que l'on souhaite obtenir.

1. $P_{2n-1}(x) = x^{2n-1} - \alpha$ avec $\alpha \neq 0$. Après avoir calculé Q_{2n-2} , l'associé de P_{2n-1} , on pose $P_{2n}(x) = x^{2n} + R(x)$. Le polynôme $R(x)$ de degré $2n - 1$ est entièrement déterminé par la relation

$$x^{2n} + R(x) = \frac{1}{Q_{2n-2}(x)}.$$

On dispose des valeurs de $R(x)$ sur un ensemble de $2n - 1$ points distincts. La valeur de α s'obtient en comparant (4.15) avec (4.11)

$$\alpha = \frac{c_{2n-1}}{c_0}.$$

2. $P_{2n-1}(x) = x^{2n-1} - \alpha x^k$ avec $k < 2n - 1$. D'après ce qui précède, on doit prendre $\alpha = c_{2n-1}/c_k$ et P_{2n} doit vérifier un système analogue à (4.14).
3. $P_{2n-1}(x) = T_{2n-1}(\alpha x)$ où T_{2n-1} désigne le polynôme de Tchebyshev de degré $2n - 1$. Cette fois on doit choisir α tel que $c(T_{2n-1}(\alpha x)) = 0$. Si on note $T_{2n-1}(x) = \sum_{i=0}^{2n-1} t_i x^i$ on a alors $\sum_{i=0}^{2n-1} c_i t_i \alpha^i = 0$ et donc α doit être une racine du polynôme $\sum_{i=0}^{2n-1} c_i t_i x^i$. On prendra de préférence la racine de module le plus proche de 1. P_{2n-1} n'est pas unitaire donc le polynôme P_{2n} récupéré par interpolation n'est pas unitaire mais fournit tout de même un polynôme orthogonal.

Choix particuliers des polynômes associés

Tout ce que nous avons fait pour P_{2n} nous pouvons l'appliquer pour choisir Q_{2n} . Par association on retrouve P_{2n-1} et par interpolation on reconstitue Q_{2n-2} . Enfin, P_{2n} s'obtient par association à partir de Q_{2n-2} . On pourrait choisir aussi le polynôme associé à l'avant-dernier polynôme orthogonal.

4.3.5 Autres formes de prolongements de la fonctionnelle

Les méthodes précédentes prolongent la fonctionnelle étudiée en lui adjoignant des moments supplémentaires après le moment d'un rang donné. Mais on peut envisager des prolongements régressifs en ajoutant des moments à la fonctionnelle avant ses premiers moments, ou des prolongements mixtes en ajoutant des moments avant et après. Une condition à remplir pour pouvoir retrouver les polynômes orthogonaux de la fonctionnelle initiale est que l'on ait suffisamment de degrés de liberté pour choisir librement le dernier polynôme orthogonal pour la fonctionnelle prolongée. Il faut veiller également à ce que, après l'adjonction de moments, la matrice de Hankel obtenue contienne comme sous-matrice la matrice de Hankel de départ. Voyons maintenant les stratégies à suivre dans ces différents cas. On désigne par $\{c_0, c_1, \dots, c_{2n+1}\}$ les moments de la fonctionnelle étudiée c .

Prolongement régressif de la fonctionnelle

On note $\{a_0, a_1, \dots, a_{2n+1}\}$ les moments qui vont devancer les moments de la fonctionnelle c et on désigne par a la fonctionnelle obtenue. La matrice de départ est enchâssée dans la matrice complétée P_r de la manière suivante.

$$P_r = \begin{pmatrix} a_0 & a_1 & \cdots & a_{n+1} & \cdots & \cdots & a_{2n} & a_{2n+1} \\ a_1 & a_2 & \cdots & a_{n+2} & \cdots & \cdots & a_{2n+1} & c_0 \\ \vdots & \vdots & & \vdots & & & \vdots & \vdots \\ a_{n+1} & a_{n+2} & \cdots & a_{2n+1} & c_0 & \cdots & c_{n-1} & c_n \\ \vdots & \vdots & & \vdots & \vdots & & \vdots & \vdots \\ a_{2n} & a_{2n+1} & \cdots & c_{n-2} & c_{n-1} & \cdots & c_{2n-2} & c_{2n-1} \\ a_{2n+1} & c_0 & \cdots & c_{n-1} & c_n & \cdots & c_{2n-1} & c_{2n} \end{pmatrix}.$$

Connaissant la fin de la fonctionnelle, l'interprétation polynomiale de la formule de Gohberg (6.9.6) nous permet d'utiliser une méthode de descente similaire. En effet, le choix laissé libre pour les $2n + 1$ premiers moments de a permet de fixer arbitrairement les zéros du polynôme de degré $2n + 1$, noté P_{2n+1} et orthogonal par rapport à la fonctionnelle a . Par construction, la fonctionnelle adjacente $a^{(2n+1)}$ a pour $2n + 1$ premiers moments ceux de c . On peut donc calculer le polynôme, noté $Q_{2n}^{(2n+1)}$, associé de P_{2n+1} par la fonctionnelle $a^{(2n+1)}$. Or la formule (6.9.6), qui généralise l'identité de Bezout aux fonctionnelles adjacentes, nous donne

$$P_{2n+1}(x)Q_{2n-1}^{(2n+1)}(x) - P_{2n}(x)Q_{2n}^{(2n+1)}(x) = x^{2n+1}.$$

Le procédé utilisé en (6.9.6) sur l'identité de Bezout peut être reproduit ici. Après avoir calculé $Q_{2n}^{(2n+1)}$ sur les $2n + 1$ racines r_i de P_{2n+1} , on déduit par inversion les valeurs de $P_{2n}(r_i)$.

$$P_{2n}(r_i) = \frac{r_i^{(2n+1)}}{Q_{2n}^{(2n+1)}(r_i)}.$$

Donc P_{2n} est entièrement déterminé par interpolation. Le polynôme de degré n orthogonal par rapport à c représente un polynôme, noté $P_n^{(2n+1)}$, orthogonal pour la fonctionnelle $a^{(2n+1)}$ adjacente à a . En remontant verticalement dans la table des polynômes orthogonaux on peut obtenir $P_n^{(2n+1)}$ à partir de P_{2n} et P_{2n+1} .

Prolongements mixtes de la fonctionnelle

On peut commencer par ajouter à la fonctionnelle c autant de moments avant qu'après. La matrice de départ reste centrée sur la diagonale de la matrice finale P_m .

$$P_{ms} = \begin{pmatrix} b_0 & b_1 & \cdots & b_{n-1} & c_0 & c_1 & \cdots & c_{n-1} & c_n \\ b_1 & b_2 & \cdots & c_0 & c_1 & c_2 & \cdots & c_n & c_{n+1} \\ \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots & \vdots \\ b_{n-1} & c_0 & \cdots & c_{n-2} & c_{n-1} & c_n & \cdots & c_{2n-2} & c_{2n-1} \\ c_0 & c_1 & \cdots & c_{n-1} & c_n & c_{n+1} & \cdots & c_{2n-1} & c_{2n} \\ c_1 & c_2 & \cdots & c_n & c_{n+1} & c_{n+2} & \cdots & c_{2n} & a_{n-1} \\ \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots & \vdots \\ c_{n-1} & c_n & \cdots & c_{2n-2} & c_{2n-1} & c_{2n} & \cdots & a_2 & a_1 \\ c_n & c_{n+1} & \cdots & c_{2n-1} & c_{2n} & a_{n-1} & \cdots & a_1 & a_0 \end{pmatrix}.$$

Si on complète de manière asymétrique, on peut faire en sorte que la matrice initiale occupe n'importe quelle position à l'intérieur de la matrice complétée.

$$P_m = \begin{pmatrix} b_0 & \cdots & b_{i-1} & c_0 & c_1 & \cdots & c_n & \cdots & c_{2n-i} \\ \vdots & & \vdots & \vdots & \vdots & & \vdots & & \vdots \\ b_{i-1} & \cdots & c_{i-2} & c_{i-1} & c_i & \cdots & c_{n+i-1} & \cdots & c_{2n-1} \\ c_0 & \cdots & c_{i-1} & c_i & c_{i+1} & \cdots & c_{n+i} & \cdots & c_{2n} \\ c_1 & \cdots & c_i & c_{i+1} & c_{i+2} & \cdots & c_{n+i+1} & \cdots & a_{2n-i-1} \\ \vdots & & \vdots & \vdots & \vdots & & \vdots & & \vdots \\ c_{n-i} & \cdots & c_{n-1} & c_n & c_{n+1} & \cdots & c_{2n} & \cdots & a_n \\ \vdots & & \vdots & \vdots & \vdots & & \vdots & & \vdots \\ c_{2n-i-1} & \cdots & c_{2n-2} & c_{2n-1} & c_{2n} & \cdots & a_{n-i-1} & \cdots & a_1 \\ c_{2n-i} & \cdots & c_{2n-1} & c_{2n} & a_{2n-i-1} & \cdots & a_{n-i} & \cdots & a_0 \end{pmatrix}.$$

Avec $i < n$ on observe que la bande constituée des moments de c est décalée vers le haut mais possède toujours une largeur de $2n + 1$ diagonales. Cela permet à la matrice initiale de s'insérer sur le bord supérieur.

4.3.6 Lien entre les approximants de type-Padé et les approximants de Padé

Replaçons ce travail dans le contexte de l'approximation rationnelle. Les approximants de type Padé ([15]) d'une série formelle

$$f(t) = \sum_{i=0}^{\infty} c_i t^i$$

sont des fonctions rationnelles qui, comme dans le cas des approximants de Padé, ont un développement en série de Taylor qui coïncide avec celui de f jusqu'à un certain ordre. Cependant, l'ordre de concordance n'est optimisé qu'après avoir choisi le dénominateur de la fraction.

Définition 3 Soit $v(t)$ un polynôme arbitraire de degré k et $w(t)$ son associé par la fonctionnelle c dont les moments sont les coefficients de la série. En supposant $c_0 \neq 0$, le polynôme $w(t)$ est de degré $k - 1$ et en notant \tilde{w} et \tilde{v} les polynômes renversés de $w(t)$ et $v(t)$ respectivement, on a

$$\frac{\tilde{w}}{\tilde{v}} - f(t) = O(t^k).$$

Une telle fraction est appelée approximant de type Padé de f et se note $(k - 1/k)_f$.

De tels approximants interviennent en accélération de la convergence. Ils forment une représentation approchée des fonctions qui tient compte des pôles et d'une partie du développement de Taylor. Lorsque l'on choisit un prolongement de la fonctionnelle pour construire un algorithme régressif, on le fait de manière implicite en choisissant un polynôme orthogonal de degré plus élevé et la fonctionnelle permet d'en déduire son associé. On construit donc un approximant de type Padé de la série formelle. Le point de départ de l'algorithme régressif est donc le choix d'un approximant de type Padé. La question qui se pose à présent est de trouver des critères de sélection d'approximants de type Padé qui permettent d'améliorer l'étape de l'algorithme d'Euclide.

4.4 Un tour d'horizon

Les deux outils qui entrent en jeu dans l'algorithme régressif et ses extensions sont l'algorithme d'Euclide et l'interpolation polynomiale. Il est donc important de préciser leurs connexions et les directions de développement que l'on peut dès lors envisager.

4.4.1 Autour de l'algorithme d'Euclide

Sans entrer dans les détails historiques, nous présentons ici et de manière non exhaustive l'évolution du cadre d'utilisation de l'algorithme d'Euclide et nous reportons son étude morphologique au paragraphe (6.9.3) qui lui est consacré. L'algorithme d'Euclide ([44], p. 129) trouve sa source dans le livre des *Eléments* (v. 300. av. J.-C.) à propos de l'étude géométrique des grandeurs commensurables. Il est réapparu sous différentes formes tout au long de sa longue histoire. De sa forme géométrique qui cherche une plus grande unité de mesure commune entre deux longueurs, se sont dégagées les applications arithmétiques, avec par exemple les équations Diophantiennes. Vu comme outil de développement itératif en fractions continues, il s'est naturellement prolongé aux polynômes. Les développements des restes intermédiaires ont permis d'obtenir des approximants rationnels, les approximants de Padé dont nous avons fait usage à plusieurs reprises tant ils sont liés aux polynômes orthogonaux. Une autre problématique s'est forgée autour de la localisation des racines d'une équation polynomiale. Avec les travaux de Sturm ([44], p. 152), qui en fournissent une élégante solution via l'algorithme d'Euclide, on observe la création d'un lien entre l'algorithme d'Euclide et un problème proche de l'interpolation. On passe donc de la recherche de racines communes à celle de la recherche de racines proprement dites.

Différents travaux récents ont focalisé mon attention sur l'algorithme d'Euclide en liaison avec les polynômes orthogonaux. Tout d'abord, il nous faut citer l'ouvrage de Draux sur la théorie des polynômes orthogonaux formels ([39]) comme nous l'avons déjà précisé, mais également les travaux de Bultheel, comme l'étude des relations entre des approximants de Padé le long d'une diagonale ([32]). Citons pour résumer un extrait du rapport de recherche ([29], p.9) où l'algorithme d'Euclide est clairement présenté comme trait d'union entre les approximants de Padé et les méthodes de Lanczos: ... we could say that the Euclidean algorithm provides a fast method for performing a Gram-Schmidt orthogonalization in an inner product space with a metric that can be indefinite and/or degenerate. It is essential however that the Gram matrix is Hankel Au niveau matriciel, il se trouve donc au centre des problèmes de type Hankel ou Toeplitz et inversement comme l'attestent ([96]) où il est fait usage de matrices de Hankel pour réaliser un algorithme d'Euclide étendu. D'autres domaines en font également une utilisation intensive, notamment en théorie de l'élimination ([67]) et dans la théorie des codes correcteurs cycliques ([4]). Mais refermons la boucle en remarquant que l'algorithme d'Euclide est aussi la charpente du théorème de Bezout et, par là même, la cheville ouvrière du théorème des restes chinois pour les méthodes modulaires. On se doute alors que les méthodes d'interpolation ne sont pas loin et la transformée de Fourier discrète s'y associe comme cas particulier. Les travaux récents portant sur les matrices polynomiales de Vandermonde, les matrices bezoutiennes et les matrices de Hankel vont dans ce sens unificateur ([68]).

Quels sont les problèmes rencontrés par l'algorithme d'Euclide? Tout d'abord le calcul numérique des restes dans les divisions successives des polynômes se heurte à deux problèmes. Si le coefficient du terme de plus haut degré d'un reste est voisin de zéro, à partir de quelle valeur numérique doit-on le considérer comme nul? De même, à partir de quel type de reste peut-on considérer que deux polynômes sont premiers entre eux ([8])? Ensuite le calcul formel a ses propres limites comme la croissance catastrophique de la taille des coefficients. Les méthodes modulaires sont dans ce cas un bon palliatif. Le fait de se ramener à l'algorithme d'Euclide laisse espérer les transferts de la plupart des résultats vers le domaine des polynômes orthogonaux et réciproquement. Nous présenterons au chapitre (6) quelques-uns de ces allers et retours. Quelles améliorations apporter à travers l'algorithme d'Euclide? On peut commencer par transposer les techniques améliorant le calcul des polynômes orthogonaux sur l'algorithme d'Euclide. Par exemple, les procédures de sauts analysées au chapitre (5) seront reformulées pour être appliquées à l'algorithme régressif. On peut aussi envisager d'appliquer des méthodes du type des moments modifiés. On peut essayer de tirer un meilleur parti de l'algorithme d'Euclide en utilisant différents procédés. Numériquement, on peut optimiser la normalisation comme le recommande ([41]) dans un algorithme composite ayant pour but la localisation de racines d'un polynôme. Enfin, on peut se ramener à un problème formel et utiliser une méthode modulaire. Dans ce cas, l'avantage indéniable de l'algorithme régressif est de produire une procédure de saut automatique prise en charge par l'algorithme d'Euclide.

4.4.2 L'interpolation et l'évaluation polynômiale

On peut considérer l'interpolation linéaire comme le point de départ de l'interpolation polynômiale. Ce sont les efforts développés pour constituer des tables numériques précises de valeurs de fonctions qui ont poussé la méthode un peu plus loin. En 1670 Gregory fait apparaître l'aspect polynômial des différences finies. Newton élargit le principe par les différences divisées. Lagrange, reprenant les travaux de Newton, décompose la difficulté sur une base de polynômes d'interpolation élémentaires et obtient une expression non itérative du polynôme d'interpolation de Newton. L'étude de l'erreur d'interpolation est entreprise par Cauchy en 1840, et le problème de convergence des interpolants est mis en évidence par Runge en 1910. Nous décrivons à présent différents aspects mathématiques de l'interpolation. Rechercher les coefficients du polynôme d'interpolation revient à inverser un système linéaire de Vandermonde, ceci pour l'aspect matriciel. Ce problème peut admettre un très mauvais conditionnement, mais la résolution est théoriquement en $O(n^2)$ pour n points d'interpolation. Il est possible d'effectuer séquentiellement l'interpolation en $O(n \log(n)^2)$ opérations ([10]).

4.4.3 La division Euclidienne et la représentation polynômiale

Suivant les informations que l'on souhaite obtenir à partir de la fonctionnelle c on peut adopter différentes représentations polynômiales. Les coefficients des polynômes orthogonaux peuvent par exemple être calculés pour différents choix de bases polynômiales. Si l'on désire calculer les racines des polynômes orthogonaux et les coefficients de Christoffel pour une méthode de quadrature de Gauss, il est alors plus judicieux de représenter les polynômes par les coefficients de la récurrence, les travaux de Gautchi ([52],[53]) montrant clairement l'intérêt de cette représentation. Dans le cas des polynômes orthogonaux classiques, le conditionnement des matrices de Hankel est en général aussi mauvais que celui d'une matrice de Hilbert, alors que le calcul des coefficients de la récurrence peut être bien conditionné. Parmi les algorithmes recensés ([59]) pour effectuer le calcul des coefficients de la récurrence, on peut citer la procédure de Stieltjes (la récurrence à trois termes), la procédure de Stieltjes discrétisée, l'algorithme de Chebyshev, l'algorithme de Chebyshev modifié et sa version discrétisée et la méthode des moments modifiés. La méthode des moments modifiés consiste à exprimer les calculs dans une autre base polynômiale afin d'obtenir un problème mieux conditionné ([95]). Comme nous l'avons constaté, l'algorithme régressif offre une grande liberté dans le choix des deux polynômes qui engendrent l'ensemble des polynômes orthogonaux par l'algorithme d'Euclide. Plusieurs questions se posent à ce stade. Quels sont parmi les choix de polynômes ceux qui, pour une fonctionnelle donnée, vont offrir un meilleur conditionnement de l'algorithme d'Euclide? Pour une fonctionnelle donnée et un choix des deux polynômes effectué, quelle est la représentation polynômiale qui favorise le plus la stabilité des calculs? Avant de proposer des éléments de réponse, il faut distinguer les représentations polynômiales adaptées à la description de l'algorithme d'Euclide.

Le degré d'un polynôme, exprimé dans une autre base que la base canonique, se détecte facilement lorsque la base est constituée avec exactement un polynôme de degré n

pour chaque $n \in \mathbb{N}$. De plus, on peut choisir une base de polynômes orthogonaux, pour faciliter les calculs. Nous allons considérer deux choix particuliers, la base de Newton et les moments modifiés.

4.4.4 La base de Newton et l'algorithme d'Euclide

Disposant d'une fonctionnelle c de moments c_0, c_1, \dots, c_{2n} , on choisit un polynôme $P_{2n}(x)$ de degré $2n$. $P_{2n}(x)$ s'exprime dans la base de Newton associée à un choix de points d'interpolation $\{x_i, i = 0, \dots, n-1\}$ de la manière suivante

$$P_{2n}(x) = \sum_{s=0}^n [x_0, \dots, x_s] (x - x_0)(x - x_1) \cdots (x - x_s),$$

où les différences divisées $[x_0, \dots, x_s]$ sont définies par l'initialisation

$$[x_i] = P_{2n}(x_i) \text{ pour } i = 0, \dots, n-1,$$

et la formule de récurrence

$$[x_i, \dots, x_j] = \frac{[x_{i+1}, \dots, x_j] - [x_i, \dots, x_{j-1}]}{x_j - x_i}. \quad (4.16)$$

L'ensemble des différences divisées $[x_i, \dots, x_j]$, relatives à P_{2n} , se calcule dans la table triangulaire des différences divisées par une récurrence sur les colonnes, les $[x_i, \dots, x_{i+k}]$ pour $i = 0, \dots, n-1-k$ composant la k -ième colonne. L'intérêt de la construction est de pouvoir effectuer les choix des points $\{x_i, i = 0, \dots, n-1\}$ au fur et à mesure des besoins et éventuellement de modifier le dernier x_i introduit s'il génère une instabilité numérique. Pour pouvoir utiliser l'identité de Bezout aux points x_i nous prenons les racines de P_{2n} pour l'ensemble des $\{x_i, i = 0, \dots, n-1\}$. On donne de préférence le polynôme $P_{2n}(x)$ par ses racines, cela permet de calculer les coefficients de P_{2n} puis le polynôme $Q_{2n-1}(x)$ associé de $P_{2n}(x)$. Enfin, on calcule les $P_{2n}(x_i)$ à partir des racines de P_{2n} et les $Q_{2n-1}(x_i)$ à partir des coefficients de Q_{2n-1} . Le polynôme P_{2n-1} orthogonal par rapport à une fonctionnelle qui prolonge c et qui admet P_{2n} pour polynôme orthogonal, peut être évalué sur l'ensemble des $\{x_i, i = 0, \dots, n-1\}$ à l'aide de l'identité de Bezout. On obtient

$$P_{2n}(x_i) = \frac{-1}{Q_{2n-1}(x_i)} \text{ pour } i = 0, \dots, n-1$$

et ce qui initialise le tableau des différences divisées et permet de calculer P_{2n} dans la base de Newton que nous avons choisie.

La deuxième étape consiste à exprimer l'algorithme d'Euclide dans la base de Newton considérée. D'après la relation de récurrence à trois termes, pour effectuer le calcul de P_{2n-2} dans la base de Newton, il faut également connaître l'expression de $xP_{2n-1}(x)$ dans notre base de Newton. La formule de Leibnitz pour la dérivation d'un produit se généralise à l'opérateur aux différences $D_{[a,b]}$ défini par

$$D_{[a,b]}(f) = \frac{f(b) - f(a)}{b - a}.$$

On a en effet

$$D_{[a,b]}(f.g) = D_{[a,b]}(f).g(a) + f(b).D_{[a,b]}(g).$$

Cet opérateur vérifie deux propriétés qui permettent d'exprimer simplement les différences divisées.

Propriété 4 Pour toute fonction f et pour tout couple de nombres complexes (a, b) tels que $a \neq b$ on a,

$$D_{[a,b]}(f) = D_{[b,a]}(f).$$

Ceci est évident et permet d'obtenir par simple développement une propriété de commutativité entre des opérateurs D particuliers.

Propriété 5 Pour toute fonction f et pour tout couple de nombres complexes (a, b) tels que $a \neq b$ on a,

$$D_{[a,x]}(D_{[b,x]}(f)) = D_{[b,x]}(D_{[a,x]}(f)) \text{ pour tout } x \text{ tel que } x \neq a \text{ et } x \neq b.$$

Relions l'opérateur D aux différences divisées. On a clairement

$$[x_0, x_1]_f = D_{[x_0, x_1]}(f),$$

puis par la première propriété de commutativité

$$\begin{aligned} [x_0, x_1, x_2]_f &= \frac{\frac{f(x_2) - f(x_1)}{x_2 - x_1} - \frac{f(x_1) - f(x_0)}{x_1 - x_0}}{x_2 - x_0} \\ &= \frac{\left(\frac{f(x) - f(x_1)}{x - x_1}\right)(x_2) - \left(\frac{f(x_1) - f(x)}{x_1 - x}\right)(x_0)}{x_2 - x_0} \\ &= \frac{(D_{[x_1, x]}(f))(x_2) - (D_{[x, x_1]}(f))(x_0)}{x_2 - x_0} \\ &= D_{[x_0, x_2]}(D_{[x, x_1]}(f)) \end{aligned}$$

La deuxième propriété de commutativité nous permet d'obtenir

$$[x_0, x_1, x_2, x_3]_f = D_{[x_0, x_3]}(D_{[x, x_2]}(D_{[x, x_1]}(f))),$$

et par récurrence on énonce

Propriété 6 Pour toute fonction f et pour tout ensemble fini de nombres complexes distincts $\{x_i, i = 0, \dots, n-1\}$ on a

$$[x_0, x_1, \dots, x_{n-1}]_f = D_{[x_0, x_{n-1}]}(D_{[x, x_{n-2}]}(\dots(D_{[x, x_0]}(f))\dots)).$$

On calcule alors les différences divisées de la fonction $xf(x)$ en utilisant la propriété (6) et la linéarité de l'opérateur D . Tout d'abord, il est clair que

$$D_{[a,b]}(xf(x)) = f(b) + a.D_{[b,a]}(f), \quad (4.17)$$

puis ensuite

$$\begin{aligned} D_{[x_0,x_2]}(D_{[x,x_1]}(xf(x))) &= D_{[x_0,x_2]}(f(x) + x_1.D_{[x,x_1]}(f)) \\ &= D_{[x_0,x_2]}(f) + x_1.D_{[x_0,x_2]}(D_{[x,x_1]}(f)) \end{aligned}$$

et ces deux égalités nous permettent d'obtenir une formulation des différences divisées de $xf(x)$ en fonction des différences divisées de $f(x)$.

Propriété 7 Pour toute fonction f et pour tout ensemble fini de nombres complexes distincts $x_i, i = 0, \dots, n-1$

$$[x_0, x_1, \dots, x_{n-1}]_{xf(x)} = [x_1, x_2, \dots, x_{n-1}]_{f(x)} + x_0.[x_0, x_1, \dots, x_{n-1}]_{f(x)}.$$

preuve:

D'après la propriété (6), on a

$$[x_0, x_1, \dots, x_{n-1}]_{xf(x)} = D_{[x_0,x_{n-1}]}(D_{[x,x_{n-2}]}(\dots(D_{[x,x_0]}(xf(x)))\dots)).$$

Au moyen de l'égalité (4.17) on substitue récursivement

$$D_{[x,x_i]}(xD_{[x,x_{i-1}]}(\dots(D_{[x,x_0]}(f(x)))\dots))$$

par

$$f(x_i) + x.D_{[x,x_i]}(\dots(D_{[x,x_0]}(f(x)))\dots)$$

car $D_{[x,x_i]}(f(x_i)) = 0$.

■

Nous remarquons que les différences divisées qui vont intervenir dans la division de P_{2n} par P_{2n-1} sont situées sur deux diagonales de la table des différences divisées de P_{2n-1} et une seule diagonale de la table de P_{2n} . Pour obtenir par récurrence les $[x_0, x_1, \dots, x_j]_{P_i}$ il nous suffit de disposer des $[x_0, x_1, \dots, x_j]_{P_{i+1}}$, $[x_1, x_2, \dots, x_j]_{P_{i+1}}$ et des $[x_0, x_1, \dots, x_j]_{P_{i+2}}$. De plus, une fois les $[x_0, x_1, \dots, x_j]_{P_i}$ calculés, on peut appliquer la définition des différences divisées (4.16) en sens inverse pour en déduire une expression des $[x_1, x_2, \dots, x_j]_{P_i}$ en fonction des $[x_0, x_1, \dots, x_j]_{P_i}$. Lorsque tous les P_i existent, on peut donc obtenir la description des P_i dans la base de Newton en ne calculant pour chacun d'eux que deux diagonales de leur table des différences divisées et cela par récurrence. Lorsqu'il y a un saut en degré, il faut calculer des diagonales en plus. Pour les polynômes $P_s(x)$ obtenus par application de l'algorithme d'Euclide à partir de P_{2n} et P_{2n-1} , si on note $\Delta_i^j(s) = [x_i, x_{i+1}, \dots, x_j]_{P_s(x)}$, la récurrence prend la forme suivante

$$\Delta_0^j(s) = \Delta_0^j(s+2) - \alpha(\Delta_1^j(s+1) + x_0\Delta_0^j(s+1) - \beta\Delta_0^j(s+1)),$$

et la construction de la deuxième diagonale provient de

$$\Delta_1^j(s) = \Delta_0^j(s) - (x_1 - x_0)\Delta_0^{j+1}(s).$$

Le coefficient α est choisi pour éliminer le coefficient en x^{s+2} entre P_{s+2} et xP_{s+1} . β permet d'éliminer le coefficient en x^{s+1} .

Dans l'algorithme d'Euclide, le calcul des restes exprimés dans une base de Newton peut s'effectuer en $O(n^2)$ opérations.

Les avantages de cette représentation proviennent du choix que l'on peut faire sur l'ordre des racines x_i dans la construction des différences divisées. Le calcul des valeurs $P_{2n-1}(x_i)$ permet de classer les valeurs x_i qui participent au mauvais conditionnement de la division Euclidienne. On veut éliminer par l'algorithme d'Euclide les différences divisées associées aux x_i qui participent le moins au mauvais conditionnement. On devra donc classer les x_i dans l'ordre des modules $|P_{2n-1}(x_i)|$ croissants. L'objectif est atteint si la moitié des divisions Euclidiennes peuvent s'opérer de manière stable, car alors on retombe sur un polynôme orthogonal par rapport à c . Cela nous permet à présent de comprendre pourquoi l'algorithme régressif sur les racines de l'unité n'améliore pas le calcul des polynômes orthogonaux par rapport à la famille des moments de Hilbert. On constate simplement que la répartition des valeurs de $P_{2n-1}(w_i)$ pour $i \in S_{2n}$ est très homogène, et donc que chaque racine de l'unité intervient de manière importante dans le mauvais conditionnement. Néanmoins, il n'est pas exclu qu'une répartition des valeurs x_i différente de S_n puisse conduire dans une autre base de polynômes à des calculs plus stables sur une partie de l'algorithme d'Euclide.

4.4.5 Méthode des moments modifiés

On exprime P_{2n} et P_{2n-1} dans une base de polynômes qui vérifient une récurrence à trois termes. L'algorithme d'Euclide reste alors en $O(n^2)$.

4.4.6 Conclusions sur l'algorithme régressif

Lorsque les premiers moments d'une fonctionnelle c sont petits ou mal calculés, on ne peut pas commencer le calcul des polynômes orthogonaux à l'aide de la récurrence à trois termes. L'algorithme régressif offre une méthode qui calcule ces polynômes en sens inverse en utilisant un prolongement fictif de la fonctionnelle c . Cette méthode peut s'interpréter comme une application de l'identité de Bezout et se décompose en une étape pour l'interpolation en $O(n^2)$ et une étape pour l'algorithme d'Euclide en $O(n^2)$. On conserve donc la complexité usuelle. Pour l'interpolation, comme pour la division il est essentiel d'essayer de maintenir éloignées les racines de P_{2n-1} et de P_{2n} . Le procédé d'interpolation de Newton permet de tenir compte de cette répartition pour aborder l'algorithme d'Euclide de manière plus efficace. Si l'instabilité de l'algorithme n'est que locale, c'est-à-dire due essentiellement à un petit nombre de racines de P_{2n} alors l'ordre utilisé dans la base de Newton permet au cours de l'algorithme d'Euclide de n'affronter les mauvais conditionnements que le plus tard possible et donc d'obtenir à partir d'un

certain rang et de manière stable des polynômes orthogonaux par rapport à la fonctionnelle de départ c . Parmi les méthodes de calcul des polynômes orthogonaux, l'algorithme régressif se situe à mi-chemin entre les méthodes de préconditionnement et les méthodes de récurrence à trois termes, sans être tout à fait une méthode de moments modifiés puisque les moments de la fonctionnelle de départ restent inchangés.

Cet algorithme révèle clairement des liens entre des outils importants. Par exemple, cette technique peut aisément s'interpréter comme une méthode de type Padé ([15]) pour le calcul des approximants de Padé. Cela permet d'envisager sous un jour nouveau les liaisons entre les algorithmes de codes correcteurs de type B.C.H. et de type Reed-Solomon.

Chapitre 5

Aspects numériques du calcul des polynômes orthogonaux

5.1 Introduction

Ce chapitre aborde le comportement numérique et l'aspect récursif du calcul des polynômes orthogonaux dans les méthodes de saut. Le premier paragraphe rappelle les principales définitions et caractéristiques qui englobent les différentes préoccupations numériques dans ce domaine. Dans le paragraphe suivant, nous illustrons les difficultés dues au calcul numérique en comparant plusieurs algorithmes usuels. Les problèmes de la régularité et de la quasi-irrégularité sont détaillés dans les paragraphes (5.4) et (5.5), cela nous conduit à présenter une forme récursive des calculs. Pour clore ce chapitre, le dernier paragraphe nous entraîne à la recherche d'une évaluation du conditionnement du problème et des questions que l'ensemble du travail soulève.

5.2 Définitions et généralités

Différents aspects des polynômes orthogonaux peuvent intervenir dans les domaines qui les utilisent. Les méthodes de quadratures de Gauss nécessitent l'obtention des racines des polynômes orthogonaux. Une méthode d'extrapolation comme l'épsilon algorithme ne requiert que les valeurs de ces polynômes pour une abscisse donnée. Enfin, seuls les coefficients de la récurrence nous intéressent lorsque l'on cherche un développement en fractions continues. Différents procédés permettent d'obtenir des informations sur les polynômes orthogonaux. On distingue principalement les méthodes de calcul des coefficients de la récurrence et les méthodes de calcul des coefficients des polynômes orthogonaux. Par exemple, le q-d algorithme ([94]) permet de construire une table des coefficients de la récurrence à trois termes qui sont aussi les coefficients de la fraction continue associée à la série formelle ou encore les composantes de la matrice tridiagonale de Jacobi (2.19). La forme progressive du q-d est la plus stable. On peut, à partir de là, construire les coefficients des polynômes orthogonaux formels ou des approximants de Padé et les zéros

des polynômes orthogonaux peuvent être obtenus par une méthode spectrale appliquée à la matrice de Jacobi. Enfin, il faut distinguer les problèmes qui sont dus à l'utilisation d'une arithmétique de précision finie. Contrairement à l'arithmétique exacte il y a parfois ambiguïté sur la notion de quantité nulle et donc sur la notion de degré d'un polynôme. Dans ce cadre là, il faut être en mesure d'évaluer la sensibilité de la solution du système aux erreurs numériques commises sur les données initiales. Cela se fait au travers de l'étude du conditionnement du système. Le conditionnement d'une matrice M est relatif à la norme matricielle choisie,

$$\text{Cond}(M) = \|M^{-1}\| \|M\|.$$

On utilise une norme matricielle induite d'une norme vectorielle. En première approximation ([58], p. 80), l'erreur relative en x pour le système $Mx = b$ peut être égale à $\text{Cond}(M)$ fois l'erreur relative en M et b . Cependant, un bon conditionnement ne garantit pas un résultat précis. Il se peut en effet que, pour l'algorithme utilisé, des opérations intermédiaires présentent un conditionnement plus mauvais que le conditionnement du système lui-même. C'est, par exemple, ce qu'il advient en général lorsque l'on n'emploie pas de stratégie de pivotage dans une factorisation LU. C'est pour cela que la méthode la plus sûre consiste à résoudre le système de Hankel (1.4) par la méthode du pivot de Gauss. On perd alors la possibilité de résoudre simultanément les systèmes intermédiaires. Pour qu'un algorithme appliqué à un système bien conditionné $Mx = b$ donne une solution calculée \tilde{x} proche de la solution exacte x on doit d'abord assurer que l'algorithme est stable ([27]), c'est-à-dire que \tilde{x} est une solution d'un système proche de la forme $(M + \tilde{M})\tilde{x} = b + \tilde{b}$, où \tilde{M} et \tilde{b} sont petits en comparaison de M et b respectivement. Si un algorithme permet de calculer des solutions de systèmes en ne commettant que de petites erreurs relatives pour une classe de matrices bien conditionnées, on dit que l'algorithme est faiblement stable pour cette classe de matrices. Par exemple, l'algorithme de Trench ([102]) pour la résolution des systèmes de Toeplitz en $O(n^2)$ n'est pas stable pour l'ensemble de toutes les matrices de Toeplitz puisqu'il échoue si un mineur principal est singulier. Par contre il est stable pour l'ensemble des matrices de Toeplitz symétriques définies positives ([27]). De nombreux autres algorithmes rapides proposés pour la résolution des systèmes de Toeplitz se sont révélés instables. Les algorithmes dans ce cas sont ceux de Durbin ([42]), Bareiss ([3]), Rissanen ([93]). Il en va de même pour les algorithmes en $O(n \log^2(n))$ de Brent-Gustavson et Yun ([13]) et de Bitmead-Anderson ([11]). Si un algorithme produit une solution quelles que soient les données de départ on le qualifie de robuste. Pour le calcul des polynômes orthogonaux formels, la robustesse d'un algorithme est sa capacité à éviter les polynômes irréguliers. L'algorithme de Trench n'est pas robuste, mais toutes les stratégies de sauts le sont bien qu'elles ne soient pas forcément stables.

5.2.1 Intérêts algorithmiques de la récurrence

Pour obtenir par exemple la valeur des polynômes orthogonaux en une abscisse fixée x_0 , la relation de récurrence s'utilise comme un schéma de Hörner. Ceci permet d'épargner

tant le développement des polynômes orthogonaux dans la base canonique que le calcul effectif des puissances de x_0 . Plus généralement, après avoir décomposé une fonction $f(x)$ dans une base de polynômes φ -orthogonaux, sous la forme

$$f(x) = \sum_{i=0}^{i=n} \varphi(f, P_i) P_i(x),$$

on peut évaluer f , en une abscisse fixée x_0 , par l'algorithme de Clenshaw ([88], p. 143). Celui-ci est également basé sur les coefficients de la récurrence. La description des polynômes orthogonaux que l'on doit adopter diffère suivant l'usage que l'on veut en faire. S'il est plus utile pour l'Analyse ou la Physique de disposer des valeurs des polynômes ou de leurs racines, il n'en est pas nécessairement de même en traitement du signal où l'information est parfois contenue dans les coefficients eux-mêmes. Observons brièvement les méthodes numériques directes utilisables pour le calcul des coefficients des polynômes orthogonaux.

Pour résoudre les systèmes (1.4), la méthode du pivot de Gauss en $O(n^3)$ ne tire aucunement parti des propriétés de la structure des matrices de Hankel. En effet, la stratégie de pivotage modifie toute la structure de la matrice et le système de taille $n+1$ ne subit pas forcément les mêmes transformations que celui de taille n . Il faut donc a priori résoudre autant de systèmes qu'il y a de polynômes orthogonaux. Néanmoins, pour des matrices symétriques définies positives, la méthode de Gauss ne nécessite plus de pivotage ([101], p. 252). De plus le second membre de chaque système de taille i est constitué, au signe près, par la colonne suivante du système de taille $i+1$ et les étapes d'éliminations peuvent être réalisées par une seule élimination de taille n (en $O(n^3)$). On gagne ainsi un facteur n sur l'élimination. Plus précisément, lorsque la matrice est fortement régulière, la méthode de Gauss conduit à la factorisation sans pivotage LDR ([101], p. 227) qui est unique. Soit

$$M_n = LDR,$$

où L et R sont des matrices triangulaires à diagonale unité, respectivement inférieure et supérieure, et D est une matrice diagonale. En utilisant la symétrie des matrices de Hankel, la factorisation prend la forme

$$M_n = LDL^T.$$

En comparant avec (1.6), la matrice triangulaire formée par les composantes des polynômes orthogonaux apparaît comme l'inverse de la matrice de factorisation précédente. En revenant au cas des matrices symétriques définies positives, on peut alors utiliser la factorisation de Cholesky ([58], p. 141) en $O(n^3)$. On obtient tout d'abord

$$M_n = B^T B,$$

avec B triangulaire inférieure. Puis on en déduit la factorisation cherchée par

$$(B^T)^{-1} M_n B^{-1} = I$$

où I désigne l'identité de $\mathbb{R}^{n \times n}$. Remarquons que la méthode de Cholesky peut s'étendre au cas non défini par une méthode par blocs ([101], p. 255). L'algorithme de Gram-Schmidt, en $O(n^3)$, n'est quant à lui pas recommandé, car il présente des tendances à la propagation d'erreurs ([92]). Par comparaison, la relation à trois termes, en $O(n^2)$, offre une méthode bien plus rapide. En outre, le calcul du polynôme orthogonal P_n ne nécessite de mémoriser que les deux précédents polynômes orthogonaux P_{n-1} et P_{n-2} . Le handicap majeur de la récurrence à trois termes est l'absence de stratégie de pivotage qui, comme dans l'algorithme du pivot de Gauss, permet de faire un choix améliorant la stabilité numérique. Dans les différentes méthodes que nous avons citées, on observe que la simultanéité de la factorisation des différents sous-systèmes se fait au détriment du pivotage. Il faut donc accepter un compromis: on compense l'absence de pivotage en sautant les polynômes dont le calcul est mal conditionné. Nous développerons ces remarques dans le chapitre (5). Du point de vue algorithmique, il est remarquable que la relation (1.7) s'interprète également en terme de division Euclidienne sur $\mathbb{C}[x]$.

5.3 Comparaison numérique d'algorithmes pour le calcul des polynômes orthogonaux

On peut observer trois sources d'erreurs numériques dans les algorithmes numériques que nous étudions. D'une part, on trouve des erreurs d'arrondi dans les données initiales, par exemple sur les moments de la fonctionnelle. D'autre part, on doit considérer les erreurs d'arrondi dues aux calculs, c'est-à-dire leur propagation. Le conditionnement du système (1.4) donne une information sur la sensibilité de la solution relativement à une perturbation initiale (le pire des cas). La propagation effective des erreurs dépend, quant à elle, de l'ordre choisi dans les calculs. Dans le cas de la récurrence, ou dans les stratégies sans pivot, c'est le conditionnement des mineurs principaux qui mesure l'instabilité interne aux calculs. Enfin, il faut tenir compte des erreurs directement imputables à l'arithmétique machine.

Exemple 14

Prenons par exemple la matrice de Hilbert d'ordre 10 pour comparer les algorithmes de Gauss, Cholesky, Gram-Schmidt et la récurrence en arithmétique à 16 chiffres de précision. La fonctionnelle associée est définie par $c_i = \frac{1}{i+1}$ pour $i = 0, 1, \dots$. La matrice des moments de dimension n présente un très mauvais conditionnement de l'ordre de $10^{1.5n}$. Ce choix permet d'observer et ce, dès les petites dimensions, de grands écarts de comportements entre les différents algorithmes utilisés, alors qu'ils conduisent tous à la solution exacte en arithmétique exacte. Pour mesurer la qualité des résultats numériques, on a calculé les erreurs relatives e_i données en norme euclidienne par la formule:

$$e_i = \frac{\|sf_i - sn_i\|_2}{\|sf_i\|_2},$$

où sf_i désigne la solution exacte calculée formellement et sn_i est le résultat numérique obtenu par un algorithme en précision finie. Le tableau ci-dessous contient les erreurs relatives e_i , à l'ordre i , produites par chaque méthode.

n	Gauss	Cholesky	Gram-Schmidt	Récurrence	Cond(M_n)
1	0	0	0	0	1
2	0	8.94427e-17	0	0	27
3	0	7.33167e-16	0	0	748
4	4.16577e-15	1.36749e-14	2.1693e-16	2.98972e-16	28375
5	7.96165e-14	4.17716e-13	6.43459e-13	5.84188e-15	943660
6	5.85275e-12	1.30108e-11	1.03626e-11	2.2739e-11	2.907e+07
7	5.40526e-12	1.80322e-10	6.34166e-10	4.68857e-10	9.852e+08
8	1.85607e-09	1.38148e-09	6.99854e-08	9.24254e-10	3.3873e+10
9	4.3085e-08	8.73015e-09	1.16748e-05	1.05288e-07	1.0997e+12
10	1.12735e-06	1.45373e-07	0.00240871	1.32679e-06	3.5357e+13
11	2.06878e-05	6.58649e-06	0.622932	0.000116609	1.2337e+15

Il apparaît clairement que l'algorithme de Gram-Schmidt est inapproprié. Les algorithmes de Gauss et Cholesky s'appuyant sur la même stratégie, donnent des résultats équivalents. La récurrence donne d'aussi bons résultats sur les premiers itérés, mais elle se montre plus sensible à la propagation des erreurs. On peut l'expliquer par le fait que, se basant uniquement sur les deux derniers itérés, les calculs ne peuvent en aucune façon compenser les dérives antérieures. Si l'on perturbe un des mineurs principaux pour le rendre mal conditionné, on met en évidence l'intérêt de la stratégie de pivotage.

Exemple 15

Prenons pour fonctionnelle $c = \{1, 0., -1, 0., 1 + \epsilon, 1, 0., -1, \dots\}$. La matrice de Jacobi à l'ordre 3 est

$$J_3 = \begin{pmatrix} 0 & 1 & 0 \\ -1 & 0 & 1 \\ 0 & -\epsilon & \epsilon^{-1} \end{pmatrix}.$$

Lorsque $\epsilon = 0$, J_3 n'existe pas et donc la relation de récurrence définissant P_3 n'existe plus. Ceci montre l'irrégularité du mineur principal d'ordre 3 lorsque $\epsilon = 0$. Pour une valeur de ϵ voisine de 0, la récurrence passe par le calcul du polynôme orthogonal de degré 3 dont le conditionnement est de l'ordre de ϵ^{-1} . On note M_n la matrice des mineurs principaux de dimension n . Avec $\epsilon = 0.333\bar{3}e-4$ et des calculs en précision 10^{-16} , on obtient les erreurs relatives suivantes

n	Récurrance	Gauss	Cond(M_n)
1	0	0	1
2	0	0	1
3	1e-11	1e-11	$\approx 12.e+4$
4	7.65151e-08	6.90895e-16	≈ 12

Cet exemple met bien en évidence l'insuffisance de la récurrence, alors que la stratégie de pivotage contourne parfaitement le mineur principal d'ordre 3.

5.4 Cas des polynômes orthogonaux non réguliers

Si un déterminant H_k s'annule, le système lié (1.4) peut soit admettre une infinité de solutions qui sont proportionnelles au polynôme orthogonal de plus haut degré qui précède, soit n'admettre aucune solution ([39], p.41). En l'occurrence, les polynômes orthogonaux réguliers vérifient encore une relation de récurrence à trois termes. Si P_k est un polynôme orthogonal régulier, alors on note par souci de simplification P_{k-1} le polynôme régulier qui le précède. Soit n_k le degré du k -ième polynôme orthogonal régulier P_k et $m_k = n_{k+1} - n_k$ le saut à effectuer, en degré, pour obtenir le polynôme orthogonal régulier suivant P_{k+1} . D'après ([23]), m_k est déterminé par les conditions suivantes

$$\begin{cases} c(x^i P_k) = 0, \text{ pour } i = 1, \dots, n_k + m_k - 2, \\ c(x^{n_k+m_k-1} P_k) \neq 0. \end{cases}$$

P_{k+1} est alors un polynôme de la forme

$$P_{k+1}(x) = w_k(x)P_k(x) - \gamma_k P_{pr(k)}(x),$$

avec $w_k(x) = a_0 + a_1 x + \dots + a_{m_k-1} x^{m_k-1} + x^{m_k}$ et

$$\gamma_k = \frac{c(x^{n_k+m_k-1} P_k)}{c(x^{n_k-1} P_{k-1})}.$$

Si on note $d_i = c(x^{i+n_k} P_k)$ et $e_i = c(x^{i+n_k} P_{k-1})$, les coefficients de w_k sont solutions du système de dimension $m_k \times m_k$

$$\begin{pmatrix} d_{m_k-1} & 0 & \dots & 0 & 0 \\ d_{m_k} & d_{m_k-1} & \dots & 0 & 0 \\ \vdots & \vdots & & \vdots & \vdots \\ d_{2m_k-3} & d_{2m_k-4} & \dots & d_{m_k-1} & 0 \\ d_{2m_k-2} & d_{2m_k-3} & \dots & d_{m_k} & d_{m_k-1} \end{pmatrix} \begin{pmatrix} a_{m_k-1} \\ a_{m_k-2} \\ \vdots \\ \vdots \\ a_0 \end{pmatrix} = \begin{pmatrix} -d_{2m_k-1} + \gamma_k e_{m_k-1} \\ -d_{m_k+1} + \gamma_k e_{m_k-2} \\ \vdots \\ \vdots \\ -d_{m_k} + \gamma_k e_0 \end{pmatrix}. \quad (5.1)$$

La régularité du système précédent est assurée par la condition $c(x^{nk+m_k-1}P_k) \neq 0$. On verra en (3.12) que ce type de système s'interprète comme un produit de polynômes modulo (x^{m_k}) ainsi que la manière de le résoudre.

Exemple 16

Prenons l'exemple (15) avec $\epsilon = 0$. On obtient par la récurrence à trois termes

$$\begin{aligned} P_0(x) &= 1, \\ P_1(x) &= x, \\ P_2(x) &= x^2 + 1. \end{aligned}$$

On observe alors que $c(x^2P_2) = 0$, mais $c(x^3P_2) \neq 0$. Donc, au rang $n_k = 2$, se présente un saut de $m_k = 2$. P_4 est calculé directement par la formule

$$P_4(x) = (a_0 + a_1x + x^2)P_2 + b_0P_1.$$

Des valeurs successives

$$\begin{aligned} c(x^3P_2) &= 1 & c(xP_1) &= -1 \\ c(x^4P_2) &= 1 & c(x^2P_1) &= 0 \\ c(x^5P_2) &= 0 & c(x^3P_1) &= 1, \end{aligned}$$

se déduit la valeur $\gamma_2 = -\frac{c(x^3P_2)}{c(xP_1)} = 1$. Alors le système (5.1) devient

$$\begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} a_1 \\ a_0 \end{pmatrix} = \begin{pmatrix} -1 \\ -1 \end{pmatrix}.$$

On en déduit $P_4(x) = (-x + x^2)P_2 + P_1 = x^2 - x^3 + x^4$.

Cette méthode s'utilise uniquement pour éviter un polynôme irrégulier. Dans la pratique, ce genre de saut se rencontre peu. En fait, la situation la plus courante est celle de polynômes quasi-irréguliers (par exemple $|\varphi(P_i, P_i)| \leq \epsilon$); les calculs sont alors mal conditionnés. Il suffit d'observer que le système (5.1) est singulier dès que l'élément diagonal d_{m_k-1} est nul, et numériquement mal conditionné lorsque ce même élément est très petit. Mais pour contourner le calcul d'un polynôme régulier correspondant à un système mal conditionné, on doit assumer des contraintes supplémentaires.

5.5 Cas des polynômes orthogonaux quasi-irréguliers.

Les polynômes quasi-irréguliers sont la principale cause de propagation d'erreurs dans la récurrence. Les méthodes proposées à l'heure actuelle pour contourner ce calcul généralisent la méthode précédente. Elles consistent à obtenir deux polynômes réguliers

successifs de manière stable à partir des deux polynômes réguliers successifs qui ont été calculés de manière stable antérieurement. Les sous-systèmes principaux intermédiaires mal conditionnés ne sont pas résolus. Un polynôme stable doit être entendu comme provenant d'un système bien conditionné; c'est là que résident une certaine liberté d'interprétation et différents choix d'exigences. Observons à partir d'un exemple l'erreur commise en considérant comme irrégulier un polynôme quasi-irrégulier plutôt qu'en procédant comme s'il était régulier.

Exemple 17

Revenons à l'exemple (15) avec $\epsilon \approx 0$ mais $0 < \epsilon$. Le polynôme P_3 est quasi-irrégulier. Si l'on prend comme approximation de P_4 le polynôme \tilde{P}_4 obtenu en traitant abusivement P_3 comme un polynôme irrégulier, on obtient d'après l'exemple (16) $\tilde{P}_4(x) = x^2 - x^3 + x^4$. Pour la fonctionnelle dont on est parti $c = \{1, 0., -1, 0., 1 + \epsilon, 1, 0., -1, \dots\}$ on constate que

$$c(\tilde{P}_4(x)) = -c(x\tilde{P}_4(x)) = c(x^2\tilde{P}_4(x)) = c(x^3\tilde{P}_4(x)) = \epsilon.$$

Le résidu est donc en norme de l'ordre de ϵ et il en est de même de l'erreur

$$e_4 = \|P_4 - \tilde{P}_4\|_2 \sim O(\epsilon).$$

◇

Cette erreur laisse entendre qu'une erreur de ϵ sur une donnée d'une étape k engendre une erreur de même ordre sur le calcul de P_k . C'est un cas bien conditionné. Cette approximation reste cependant inexploitable en général dans le calcul des polynômes orthogonaux. L'erreur immanquablement engendrée par cet abus peut se propager de manière catastrophique sur les calculs des polynômes orthogonaux suivants. Exposons un tel exemple.

Exemple 18

Pour la fonctionnelle c de moments $\{1, 1, 1 + 0.000000001, 0, 0, 100000, -1, 1, 1, 1, 1, \dots\}$, nous calculons les polynômes orthogonaux par la récurrence à trois termes avec une arithmétique à 16 chiffres de précision

$$\begin{aligned} P_0 &= 1., \\ P_1 &= t - 1., \\ P_2 &= t^2 + 10000000000001.0t - 10000000000002.0, \\ P_3 &= t^3, \\ P_4 &= t^4 + 0.99999999t^3 - 9.9999999997 \times 10^{-14}t^2 - 0.999999999998t \\ &\quad + 0.999999999999, \\ P_5 &= t^5 + 1.00004999099992t^4 - 9997.999949989t^3 - 0.999999999998t^2 \\ &\quad + 0.99994999899998t + 0.000050000999919995, \end{aligned}$$

puis par une récurrence qui saute le polynôme irrégulier de degré 2, en ayant considéré la fonctionnelle comme irrégulière.

$$\begin{aligned}
P_0 &= 1, \\
P_1 &= t - 1.0, \\
P_3 &= -100000.0 + t^3 + 100000.0t, \\
P_4 &= 999999999999999.0t + t^4 + 100000.0t^2 - 1000000000099999.0 + 10000000001.0t^3, \\
P_5 &= t^5 + 1.000010t^4.
\end{aligned}$$

Pour obtenir les polynôme solutions, on a remplacé le coefficient $1 + 0.000000001$ par un $1 + \epsilon$ et dans les résultats de la récurrence à trois termes obtenus formellement on a ensuite effectué une substitution $\epsilon = 0.000000001$. Cela nous donne finalement

$$\begin{aligned}
P_3 &= t^3 + 0.0000999999997t^2 + 99999.9998t - 99999.9999, \\
P_4 &= -11111111148148.1t - 11111.11113703700t^2 - 1111111112.703693t^3 \\
&\quad + t^4 + 11111111159259.2, \\
P_5 &= 0.1140700386t - 0.00001000010055555000t^2 + 1.000001740829630t^3 \\
&\quad + 1.000009999899998t^4 + t^5 - 100000.10000000000.
\end{aligned}$$

On constate alors que les des deux méthodes précédentes ne conviennent pas.

◇

La récurrence à mettre en oeuvre dans le cas des polynômes quasi-irréguliers est légèrement différente. D'après la récurrence à trois termes, P_{k+1} est dans l'idéal engendré par P_k et P_{k-1} . Ainsi P_{k+2} , qui est dans l'idéal engendré par P_{k+1} et P_k , se trouve également dans l'idéal engendré par P_k et P_{k-1} . En regard de la remarque précédente et sur base du respect des degrés, on est amené à poser comme dans ([25])

$$P_{k+1}(x) = w_k(x)P_k(x) + v_k(x)P_{k-1}(x), \quad (5.2)$$

avec $w_k(x) = a_0 + a_1x + \dots + a_{m_k-1}x^{m_k-1} + a_{m_k}x^{m_k}$ et $v_k(x) = b_0 + b_1x + \dots + b_{m_k-1}x^{m_k-1}$.

Supposons que $\deg(P_{k+1}) = n_k + m_k$, $\deg(P_k) = n_k$ et $\deg(P_{k-1}) = n_k - 1$. On tient compte ici d'un saut éventuel de taille m_k entre P_k et P_{k+1} . Pour obtenir à partir de P_k un polynôme de degré $n_k + m_k$ il est nécessaire de choisir $\deg(w_k) \geq m_k$. Afin de minimiser le nombre d'inconnues, on prend $\deg(w_k) = m_k$ et on constatera ensuite que ce choix est impératif pour faire coïncider le nombre d'inconnues avec le nombre d'équations. Pour des raisons identiques, on est amené à poser $\deg(v_k) \leq m_k - 1$. On considère tout d'abord que m_k est petit par rapport à n_k ou plus précisément que $m_k \leq n_k$. Cette dernière condition assure que la première contrainte $c(P_{k+1})$ est immédiatement satisfaite ($n_k > m_k$) ou qu'elle ne fait intervenir qu'une inconnue ($m_k = n_k$). On utilise alors les propriétés d'orthogonalité pour observer que

$$c(x^i w_k P_{n_k}) = 0 \text{ pour } i = 0, \dots, n_k - m_k - 1$$

et que

$$c(x^i w_k P_{n_k}) = 0 \text{ pour } i = 0, \dots, n_k - m_k - 1.$$

Détaillons les $2m_k$ conditions d'orthogonalité sur P_{k+1} utilisables et qui correspondent au choix de $i = n_k - m_k$ jusqu'à $i = n_k + m_k - 1$. La première condition d'orthogonalité non triviale se manifeste pour $i = n_k - m_k$ et prend la forme

$$\begin{aligned} 0 &= c(x^{n_k - m_k} P_{k+1}) \\ &= a_0 c(x^{n_k - m_k} P_k) + \dots + a_{m_k} c(x^{n_k} P_k) + b_0 c(x^{n_k - m_k} P_{k-1}) + \dots + b_{m_k - 1} c(x^{n_k - 1} P_{k-1}). \end{aligned}$$

Pour respecter la normalisation unitaire on a $a_{m_k} = 1$. Seule la dernière inconnue $b_{m_k - 1}$ dispose d'un coefficient non nul $c(x^{n_k - 1} P_{k-1})$ ce qui fournit

$$0 = c(x^{n_k} P_k) + b_{m_k - 1} c(x^{n_k - 1} P_{k-1}).$$

La condition suivante obtenue pour $i = n_k - m_k + 1$ fait intervenir deux inconnues de plus

$$0 = c(x^{n_k - m_k + 1} P_{k+1}) = a_{m_k - 1} c(x^{n_k} P_k) + c(x^{n_k + 1} P_k) + b_{m_k - 2} c(x^{n_k - 1} P_{k-1}) + b_{m_k - 1} c(x^{n_k} P_{k-1}).$$

Chaque condition introduit deux inconnues supplémentaires jusqu'au rang $i = n_k - 1$

$$\begin{aligned} 0 = c(x^{n_k - 1} P_{k+1}) &= a_1 c(x^{n_k} P_k) + \dots + a_{m_k - 1} c(x^{n_k + m_k - 2} P_k) + c(x^{n_k + m_k - 1} P_k) \\ &\quad + b_0 c(x^{n_k - 1} P_{k-1}) + \dots + b_{m_k - 1} c(x^{n_k + m_k - 2} P_{k-1}). \end{aligned}$$

La condition pour $i = n_k$ introduit la dernière inconnue a_0 et décale encore les coefficients comme suit

$$\begin{aligned} 0 = c(x^{n_k} P_{k+1}) &= a_0 c(x^{n_k} P_k) + \dots + a_{m_k - 1} c(x^{n_k + m_k - 1} P_k) + c(x^{n_k + m_k} P_k) \\ &\quad + b_0 c(x^{n_k} P_{k-1}) + \dots + b_{m_k - 1} c(x^{n_k + m_k - 1} P_{k-1}). \end{aligned}$$

Jusqu'au rang $i = n_k + m_k - 1$ les conditions suivantes ne font que décaler les coefficients

$$\begin{aligned} 0 = c(x^{n_k + m_k - 1} P_{k+1}) &= a_0 c(x^{n_k + m_k - 1} P_k) + \dots + a_{m_k - 1} c(x^{n_k + 2m_k - 2} P_k) + c(x^{n_k + 2m_k - 1} P_k) \\ &\quad + b_0 c(x^{n_k + m_k - 1} P_{k-1}) + \dots + b_{m_k - 1} c(x^{n_k + 2m_k - 2} P_{k-1}). \end{aligned}$$

A priori, la résolution de ce système linéaire nécessite l'emploi d'une méthode classique comme le suggère ([23]). Comparativement au procédé de récurrence, chaque tentative de saut coûte donc très cher en temps de calcul, d'autant qu'on ne sait pas à l'avance le nombre de sauts et leurs tailles. En arrangeant convenablement le système d'équations on peut cependant le présenter sous la forme d'un problème de recherche d'approximants de Padé, le résoudre par une méthode de bordage ou encore l'aborder comme un système de Hankel de type (1.4). Commençons par observer l'organisation des coefficients en présentant une écriture matricielle des contraintes linéaires précédentes. Nous posons $q_i = c(x^{n_k + i} P_{n_k})$ pour $i = 0, \dots, 2m_k - 1$ et $p_i = c(x^{n_k - 1 + i} P_{n_k - 1})$ pour $i = 0, \dots, 2m_k - 2$ et nous utilisons (3.11) afin de reconnaître l'écriture matricielle de produits de polynômes

tronqués à un certain ordre.

$$\begin{pmatrix} q_0 & 0 & \cdots & 0 \\ q_1 & q_0 & 0 & \vdots \\ \vdots & \vdots & & \vdots \\ q_{m_k-2} & \cdots & 0 & 0 \\ q_{m_k-1} & \cdots & q_0 & 0 \\ q_{m_k} & q_{m_k-1} & \cdots & q_0 \\ q_{m_k+1} & q_{m_k} & \cdots & q_1 \\ \vdots & \vdots & & \vdots \\ q_{2m_k-2} & q_{2m_k-3} & \cdots & q_{m_k-2} \\ q_{2m_k-1} & q_{2m_k-2} & \cdots & q_{m_k-1} \end{pmatrix} \begin{pmatrix} 1 \\ a_{m_k-1} \\ \vdots \\ a_1 \\ a_0 \end{pmatrix} + \begin{pmatrix} p_0 & 0 & \cdots & 0 \\ p_1 & p_0 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ p_{m_k-2} & \cdots & p_0 & 0 \\ p_{m_k-1} & p_{m_k-2} & \cdots & p_0 \\ p_{m_k} & p_{m_k-1} & \cdots & p_1 \\ p_{m_k+1} & p_{m_k} & \cdots & p_2 \\ \vdots & \vdots & & \vdots \\ p_{2m_k-2} & p_{2m_k-3} & \cdots & p_{m_k-2} \\ p_{2m_k-1} & p_{2m_k-2} & \cdots & p_{m_k-1} \end{pmatrix} \begin{pmatrix} b_{m_k-1} \\ \vdots \\ b_1 \\ b_0 \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \quad (5.3)$$

En notant $Q(x) = q_0 + q_1x + \cdots + q_{2m_k-1}x^{2m_k-1}$ et $P(x) = p_0 + p_1x + \cdots + p_{2m_k-1}x^{2m_k-1}$ on est amené à chercher deux polynômes $\tilde{a}(x) = 1 + a_{m_k-1}x + \cdots + a_0x^{m_k}$ et $\tilde{b}(x) = b_{m_k-1} + b_{m_k-2}x + \cdots + b_0x^{m_k-1}$ vérifiant

$$Q(x)\tilde{a}(x) + P(x)\tilde{b}(x) = O(x^{2m_k}). \quad (5.4)$$

Par ailleurs, trouver un approximant de Padé (1.9) de type $[n/m]$ pour la fonction rationnelle $f(x) = P(x)/Q(x)$ consiste à construire deux polynômes $b(x)$ et $a(x)$, de degrés inférieurs à m et n respectivement, vérifiant l'égalité

$$b(x)\frac{P(x)}{Q(x)} - a(x) = O(x^{n+m+1}). \quad (5.5)$$

En multipliant (5.5) par $Q(x)$ et en comparant à (5.4), les choix de $n = m_k$ et $m = m_k - 1$ permettent d'identifier $-a(x)/b(x)$ comme l'approximant de Padé $[m_k/m_k - 1]$ de $f(x)$ lorsque celui-ci existe. A ce stade, il ne faut pas perdre de vue que la recherche d'un approximant de Padé ou la recherche d'un polynôme orthogonal sont deux problèmes identiques puisque basés sur la résolution de systèmes de Hankel de même forme. On peut donc aisément convertir l'aspect du problème d'une forme à l'autre. Vu sous l'angle des polynômes orthogonaux, on obtient le théorème suivant.

Théorème 2

Dans les combinaisons linéaires de deux polynômes orthogonaux réguliers consécutifs de degrés $n_k - 1$ et n_k respectivement qui permettent d'exprimer directement les polynômes orthogonaux suivants jusqu'à un degré $m_k = n_k$, les deux familles de coefficients polynômiaux forment, en les renversant, deux familles de polynômes orthogonaux.

preuve:

La première étape consiste à montrer que les coefficients successifs constituent des approximants de Padé successifs d'une même fonction $f(x)$. D'après nos remarques, chaque couple $(a_{m_k}(x), b_{m_k-1}(x))$ de coefficients associé au saut de taille m_k décrit un approximant de Padé $[m_k/m_k - 1]$ d'une fonction rationnelle $P_{2m_k-1}(x)/Q_{2m_k-1}(x)$. Lorsque l'on passe d'un saut de taille m_k au saut de taille $m_k + 1$, on augmente de deux le nombre d'inconnues, le nombre d'équations et donc aussi les degrés respectifs de P et Q . Par construction, les coefficients de P_{2m_k+1} et Q_{2m_k+1} sont obtenus en ajoutant deux monômes à P_{2m_k-1} et Q_{2m_k-1} respectivement.

$$Q_{2m_k+1}(x) = Q_{2m_k-1}(x) + q_{2m_k}x^{2m_k} + q_{2m_k+1}x^{2m_k+1},$$

$$P_{2m_k+1}(x) = P_{2m_k-1}(x) + p_{2m_k}x^{2m_k} + p_{2m_k+1}x^{2m_k+1}.$$

Partant de l'égalité

$$\begin{aligned} Q_{2m_k+1}(x)a_{m_k}(x) + P_{2m_k+1}(x)b_{m_k-1}(x) = \\ Q_{2m_k-1}(x)a_{m_k}(x) + P_{2m_k-1}(x)b_{m_k-1}(x) \\ + (q_{2m_k}x^{2m_k} + q_{2m_k+1}x^{2m_k+1})a_{m_k}(x) + (p_{2m_k}x^{2m_k} + p_{2m_k+1}x^{2m_k+1})b_{m_k-1}(x), \end{aligned}$$

puisque l'on a

$$Q_{2m_k-1}(x)a_{m_k}(x) + P_{2m_k-1}(x)b_{m_k-1}(x) = O(x^{2m_k}),$$

cela assure que

$$Q_{2m_k+1}(x)a_{m_k}(x) + P_{2m_k+1}(x)b_{m_k-1}(x) = O(x^{2m_k}).$$

Autrement dit, le couple $(a_{m_k}(x), b_{m_k-1}(x))$ forme un approximant de Padé $[m_k/m_k - 1]$ de $P_{2m_k+1}(x)/Q_{2m_k+1}(x)$. Le résultat se prolonge tout aussi simplement aux fractions $P_{2m_k+i}(x)/Q_{2m_k+i}(x)$ pour $i \in \mathbb{N}$.

La deuxième étape consiste à appliquer des résultats, exposés dans ([16]), sur les liaisons entre les dénominateurs des approximants de Padé et les polynômes orthogonaux.

Propriété 8 Soit $f(x)$ une fonction dont le développement de Taylor à l'origine est $\sum_{i=0}^{\infty} c_i x^i$. L'approximant de Padé $[n - 1/n]_f$ de $f(x)$, lorsqu'il existe, est donné par

$$[n - 1/n]_f(x) = \frac{\tilde{Q}_n(x)}{\tilde{P}_n(x)},$$

où les \tilde{P}_n sont les polynômes orthogonaux de degré n par rapport à la fonctionnelle de moments $(c_i)_{i \in \mathbb{N}}$ et les \tilde{Q}_n sont leurs associés (3.2.1).

D'après cette propriété les polynômes $b_{m_k-1}(x)$ sont orthogonaux par rapport à la fonctionnelle de moments $(c_i)_{i \in \mathbb{N}}$ définie par le développement de Taylor de $f(x)$. Quant aux polynômes associés $a_{m_k}(x)$, ils sont également orthogonaux ([14]) mais par rapport à la fonctionnelle de moments $(c_{i+2})_{i \in \mathbb{N}}$.

■

Ce résultat permet donc d'envisager de nouvelles stratégies pour le calcul des polynômes orthogonaux en appliquant pour les sauts des méthodes récurrentes équivalentes à celles rencontrées dans le cas régulier. Néanmoins, la situation diffère par la forme des données de départ. Pour le calcul des coefficients polynomiaux qui interviennent dans les sauts, on ne dispose pas directement de la fonctionnelle. Les polynômes intermédiaires, dont le calcul présente un mauvais conditionnement, ne sont plus calculés mais donnés par des combinaisons linéaires à partir de deux polynômes orthogonaux précédents dont le calcul présente une perte de précision acceptable. Remarquons que les premiers moments de la fonctionnelle définie à partir du développement de Taylor de la fonction f et qui intervient dans le saut, ne permettent pas un calcul précis des polynômes orthogonaux par une relation de récurrence. En effet, les premiers p_i et q_i peuvent être très petits à cause de la quasi-irrégularité. De plus le mauvais conditionnement des polynômes dont on cherche à éviter le calcul se reporte sur les polynômes \tilde{a} et \tilde{b} qui servent de composantes dans le saut (5.3,5.2). Il ne faut donc pas utiliser une stratégie progressive. Une stratégie du type de l'algorithme régressif est donc toute indiquée dans ce cas si on parvient à compléter la fonctionnelle de telle manière qu'il n'y ait pas de mauvais conditionnement par au-dessus.

On pourrait également proposer une généralisation du procédé en l'appliquant aux techniques de contournement des polynômes irréguliers ou mal conditionnés utilisant les familles de polynômes adjacents et utiliser des relations de récurrences et méthodes de déplacement dans les tables de Padé que l'on trouve dans les travaux de Draux et Van Ingelandt ([40]).

On ne peut pas tenir le même raisonnement si P_k et P_{k-1} sont deux polynômes réguliers stables mais non nécessairement consécutifs. En effet, si l'idéal de $\mathbb{C}[x]$ engendré par deux polynômes orthogonaux réguliers consécutifs contient tous les polynômes orthogonaux réguliers de degré supérieur il n'en va pas de même lorsque l'on prend des polynômes orthogonaux non successifs comme le montre l'exemple suivant

Exemple 19

Les premiers polynômes orthogonaux par rapport à la fonctionnelle

$$c = \{1, 0, 1, 0, -2, 0, 1, -3, 1, 9\}$$

sont

$$\begin{aligned}
P_0(x) &= 1, \\
P_1(x) &= 3x, \\
P_2(x) &= x^2 - 1, \\
P_3(x) &= x^3 + 2x, \\
P_4(x) &= (x-1)P_3(x) - P_2(x) = x^4 - x^3 + x^2 - 2x + 1, \\
P_5(x) &= (x+1)P_4(x) + P_3(x) = x^5 + x^3 - x^2 + x + 1.
\end{aligned}$$

Les polynômes P_2 et P_4 ont été construits de manière à avoir une racine commune $x = 1$. Donc tout polynôme de l'idéal $I(P_2, P_4)$ engendré par P_2 et P_4 admet $x = 1$ comme racine. Or $P_5(1) = P_3(1) = 3$ donc $P_5 \notin I(P_2, P_4)$ et P_5 ne peut pas être obtenu à partir d'une combinaison linéaire de P_2 et P_4 à coefficients polynomiaux.

De nombreux algorithmes sont basés sur les méthodes de saut ([24],[39]). Ce faisant, on introduit deux nouvelles difficultés. D'une part, les sauts doivent rester de petite taille, puisque chaque saut réclame la résolution du système (6.9.6). D'autre part, les critères de décision de saut nécessitent une évaluation du conditionnement du système (1.4) et ce sujet fait encore l'objet de nombreuses recherches ([5]). On peut, par exemple, remarquer que la condition $\varphi(P_i, P_i) \leq \epsilon$ n'est pas la seule source de mauvais conditionnement.

Exemple 20

Dans l'exemple suivant, prenons la fonctionnelle $c = \{1, \sqrt{\epsilon}, 0, 1 - \epsilon, 2 - \frac{1}{\epsilon}, 0\}$ qui admet pour polynômes orthogonaux successifs

$$\begin{aligned}
P_0(x) &= 1, \\
P_1(x) &= -\sqrt{\epsilon} + x, \\
P_2(x) &= \sqrt{\epsilon} - \frac{1}{\sqrt{\epsilon}} + \left(\frac{1-\epsilon}{\epsilon}\right)x + x^2, \\
P_3(x) &= O(\epsilon^{-7/2}) + (O(\epsilon^{-4}))x + (\epsilon^{-3} + O(\epsilon^{-2}))x^2 + x^3,
\end{aligned}$$

où les coefficients de P_3 ont été développés suivant ϵ au voisinage de zéro. On peut constater le comportement catastrophique en $O(\epsilon^{-4})$ du coefficient du monôme de degré un dans le polynôme P_3 . Pourtant, les valeurs $\varphi(P_1, P_1) = -\epsilon$ et $\varphi(P_2, P_2) = O(\epsilon)$ ne laissent pas présager un comportement aussi mauvais. Les conditionnements des matrices de Hankel M_2 et M_3 sont respectivement

$$\text{cond}(M_2) = O(\epsilon^{-1}), \text{cond}(M_3) = O(\epsilon^{-4}).$$

En examinant les ϵ qui contribuent au mauvais conditionnement, on trouve la répartition suivante

$$\varphi(P_2, P_2) = O(\epsilon^{-1}) \quad \varphi(P_2, x^3) = O(\epsilon^2).$$

Autrement dit, le coefficient α_3 de la relation de récurrence (1.8) est en $O(\epsilon^{-3})$. Ce coefficient conjugué avec le coefficient de P_2 en $O(\epsilon^{-1})$ explique le mauvais conditionnement pour le calcul de P_3 . Parallèlement, en obtenant une croissance des coefficients comparable au conditionnement, cet exemple nous révèle que le conditionnement $\text{cond}(M_n)$ n'est

pas une surestimation des difficultés envisageables. Pour des considérations de conditionnement, on pourra également apprécier l'exemple suivant extrait de ([12], p. 150).

Exemple 21

On étudie la fonctionnelle $c = \{-\gamma, 1, 0, 1, 1 + \gamma, 0\}$ et l'on choisit un nombre γ assez grand. Les polynômes orthogonaux par rapport à c sont

$$\begin{aligned} P_0(x) &= 1; \\ P_1(x) &= \frac{1}{\gamma} + x, \\ P_2(x) &= -1 - \gamma x + x^2. \end{aligned}$$

Le conditionnement de M_3 est $\text{cond}(M_3) = (\gamma + 2)(\gamma + 3)$. On obtient donc un mauvais conditionnement sans que le coefficient $c(P_2^2) = 1$ le laisse paraître. Par contre, le carré de la norme du vecteur étant très grand, cela conduit à l'instabilité.

La relation de récurrence doit donc subir les conditionnements successifs des mineurs principaux. Dans le cas d'un saut, seuls serviront les mineurs correspondants aux deux derniers polynômes orthogonaux calculés. En s'appuyant sur l'exemple (15), on peut observer que le conditionnement des mineurs principaux n'est pas nécessairement croissant avec la dimension, ce qui justifie l'intérêt des sauts. Pour mettre en œuvre une stratégie de sauts, on notera donc l'importance d'une évaluation du conditionnement et de maîtriser un certain nombre de paramètres qui l'affectent. Les deux exemples précédents, mettent en évidence plusieurs sources d'instabilité avec leur répartition.

5.6 Conditionnement des polynômes orthogonaux

Il est important de mesurer le conditionnement d'une matrice de Hankel quelconque pour évaluer la sensibilité aux erreurs d'arrondi des polynômes orthogonaux formels calculés en arithmétique de précision finie. Il faut en effet pouvoir déterminer si un polynôme orthogonal déjà calculé peut raisonnablement servir de point de départ au calcul des suivants. Reprenons les notations utilisées en (2.2). M_k et M_{k+1} désignent les matrices de Hankel utilisées pour le calcul des deux polynômes orthogonaux unitaires successifs P_k et P_{k+1} . La matrice M_{k+1} est reconstituée à partir de M_k au moyen du bordage suivant

$$M_{k+1} = \begin{pmatrix} M_k & u_k \\ u_k^T & c_{2k} \end{pmatrix}$$

avec $u_k^T = (c_k, c_{k+1}, \dots, c_{2k-1})$. Pour compléter le point de vue matriciel, on note indifféremment le polynôme P_k et le vecteur dont les composantes sont les coefficients de P_k donnés dans l'ordre croissant, $P_k = (p_0, p_1, \dots, p_{k-1}, 1)^T$. La restriction du vecteur P_k à ses k premières composantes est notée $s_k = (p_0, p_1, \dots, p_{k-1})^T$. L'équation (1.4) se traduit alors, au choix, par l'une des égalités

$$M_k s_k = -u_k$$

ou

$$M_{k+1}P_k = \begin{pmatrix} o_k \\ c(P_k^2) \end{pmatrix},$$

où o_k est le vecteur nul de dimension k . En suivant le principe de factorisation LDL^T d'une matrice symétrique ([101], p.248), on obtient successivement

$$\begin{aligned} M_{k+1} &= \begin{pmatrix} M_k & u_k \\ u_k^T & c_{2k} \end{pmatrix} \\ &= \begin{pmatrix} M_k & o_k \\ u_k^T & c_{2k} + u_k^T s_k \end{pmatrix} \begin{pmatrix} I_k & -s_k \\ o_k^T & 1 \end{pmatrix} \\ &= \begin{pmatrix} I_k & o_k \\ -s_k^T & 1 \end{pmatrix} \begin{pmatrix} M_k & o_k \\ o_k^T & c_{2k} + u_k^T s_k \end{pmatrix} \begin{pmatrix} I_k & -s_k \\ o_k^T & 1 \end{pmatrix} \\ &= \begin{pmatrix} O_k & o_k \\ o_k^T & c_{2k} + u_k^T s_k \end{pmatrix} + \begin{pmatrix} I_k & o_k \\ -s_k^T & 0 \end{pmatrix} \begin{pmatrix} M_k & o_k \\ o_k^T & s_k \end{pmatrix} \begin{pmatrix} I_k & -s_k \\ o_k & 0 \end{pmatrix}. \end{aligned} \quad (5.6)$$

O_k et I_k désignent respectivement les matrices nulle et unité de dimension k . En utilisant la norme matricielle induite par la norme euclidienne, on a

$$\left\| \begin{pmatrix} O_k & o_k \\ o_k^T & 1 \end{pmatrix} \right\|_2 = 1,$$

ce qui impose

$$\begin{aligned} \left\| \begin{pmatrix} O_k & o_k \\ o_k^T & c_{2k} + u_k^T s_k \end{pmatrix} \right\|_2 &= |c_{2k} + u_k^T s_k|, \\ \left\| \begin{pmatrix} M_k & o_k \\ o_k^T & 0 \end{pmatrix} \right\|_2 &= \|M_k\|_2, \end{aligned}$$

puis enfin,

$$\left\| \begin{pmatrix} I_k & o_k \\ s_k^T & 0 \end{pmatrix} \right\|_2 \leq \|P_k\|_2. \quad (5.7)$$

Détaillons cette dernière inégalité. Pour tout vecteur $x_k \in \mathbb{C}^k$ et $\alpha \in \mathbb{C}$ tels que $\|(x_k, \alpha)^T\|_2 = 1$, on obtient

$$\left\| \begin{pmatrix} I_k & o_k \\ s_k^T & 0 \end{pmatrix} \begin{pmatrix} x_k \\ \alpha \end{pmatrix} \right\|_2 = \left\| \begin{pmatrix} x_k \\ -s_k^T x_k \end{pmatrix} \right\|_2.$$

Le résultat ne dépendant pas de α , on a les majorations $\|x_k\|_2 \leq 1$ et

$$\left\| \begin{pmatrix} x_k \\ -s_k^T x_k \end{pmatrix} \right\|_2^2 = \|x_k\|_2^2 + |s_k^T x_k|^2 \leq 1 + |s_k^T x_k|^2. \quad (5.8)$$

D'après l'inégalité de Cauchy-Schwarz ([70], p.261) nous pouvons déduire

$$1 + |s_k^T x_k|^2 \leq 1 + \|s_k\|_2^2 \|x_k\|_2^2 \leq 1 + \|s_k\|_2^2. \quad (5.9)$$

Pour compléter l'inégalité (5.9) remarquons que

$$\|P_k\|_2^2 = \|(s_k, 1)\|_2^2 = 1 + \|s_k\|_2^2. \quad (5.10)$$

En revenant à la définition de la norme matricielle, cela donne

$$\left\| \begin{pmatrix} I_k & o_k \\ s_k^T & 0 \end{pmatrix} \right\|_2 = \sup_{\|x_k\|_2 \leq 1} \left\| \begin{pmatrix} I_k & o_k \\ s_k^T & 0 \end{pmatrix} \begin{pmatrix} x_k \\ \alpha \end{pmatrix} \right\|_2 \leq \|P_k\|_2,$$

et ceci, en utilisant successivement les inégalités (5.8) et (5.9), puis l'égalité (5.10).

De (5.6) et (5.7) se déduit une majoration de la norme de M_{k+1} , à savoir

$$\|M_{k+1}\|_2 \leq |c_{2k} + u_k^T s_k| + \|P_k\|_2^2 \|M_k\|_2.$$

Par le biais de manipulations similaires, nous pouvons également majorer la norme de M_{k+1}^{-1} . Partant de l'égalité (5.6), on trouve

$$\begin{aligned} M_{k+1}^{-1} &= \begin{pmatrix} I_k & -s_k \\ o_k^T & 1 \end{pmatrix}^{-1} \begin{pmatrix} M_k & o_k \\ o_k^T & c_{2k} + u_k^T s_k \end{pmatrix}^{-1} \begin{pmatrix} I_k & o_k \\ -s_k^T & 1 \end{pmatrix}^{-1} \\ &= \begin{pmatrix} I_k & s_k \\ o_k^T & 1 \end{pmatrix} \begin{pmatrix} M_k^{-1} & o_k \\ o_k^T & \frac{1}{c_{2k} + u_k^T s_k} \end{pmatrix} \begin{pmatrix} I_k & o_k \\ s_k^T & 1 \end{pmatrix} \\ &= \begin{pmatrix} M_k^{-1} & o_k \\ o_k^T & 0 \end{pmatrix} + \frac{1}{c_{2k} + u_k^T s_k} \begin{pmatrix} I_k & s_k \\ o_k & 1 \end{pmatrix} \begin{pmatrix} I_k & o_k \\ s_k^T & 1 \end{pmatrix}. \end{aligned}$$

Il en découle la majoration suivante

$$\|M_{k+1}^{-1}\|_2 \leq \|M_k^{-1}\|_2 + \frac{1}{|c_{2k} + u_k^T s_k|} \|P_k\|_2^2.$$

En assemblant les majorations des normes de M_{k+1}^{-1} et M_{k+1} et en notant $\gamma = c_{2k} + u_k^T s_k$, il apparaît que

$$\begin{aligned} \|M_{k+1}^{-1}\|_2 \|M_{k+1}\|_2 &\leq \left(\|M_k^{-1}\|_2 + \frac{1}{|\gamma|} \|P_k\|_2^2 \right) (|\gamma| + \|P_k\|_2^2 \|M_k\|_2) \\ &\leq \|P_k\|_2^2 \left(\|M_k^{-1}\|_2 \|M_k\|_2 + \frac{|\gamma|}{\|P_k\|_2^2} \|M_k^{-1}\|_2 + \frac{\|P_k\|_2^2}{|\gamma|} \|M_k\|_2 + 1 \right). \end{aligned}$$

En remarquant que $\gamma = c(x^k P_k) = c(P_k^2)$, ce qui précède démontre la propriété suivante.

Propriété 9

Le conditionnement d'une matrice de Hankel M_{k+1} , estimé à partir de son plus grand mineur principal M_k , vérifie avec les notations antérieures

$$\text{cond}_2(M_{k+1}) \leq \|P_k\|_2^2 \left(\text{cond}_2(M_k) + \frac{|c(P_k^2)|}{\|P_k\|_2^2} \|M_k^{-1}\|_2 + \frac{\|P_k\|_2^2}{|c(P_k^2)|} \|M_k\|_2 + 1 \right).$$

Au travers de cette inégalité, il est possible d'observer comment les coefficients $|c(P_k^2)|$ et $\|P_k\|_2^2$ interviennent pour contrebalancer l'évolution du conditionnement. Cependant, ne concernant que le conditionnement de deux mineurs consécutifs, ce résultat demeure peu utilisable dès qu'une stratégie de sauts est mise en place. En outre, P_k étant unitaire, on a nécessairement $\|P_k\| \geq 1$ ce qui occulte les diminutions éventuelles de conditionnement. La majoration obtenue ne traduit donc pas toutes les variations du conditionnement, mais elle prévoit cependant les cas catastrophiques. Une minoration du conditionnement offrirait un meilleur contrôle pour effectuer les sauts dans les méthodes de bordage.

5.7 Conclusions

La récursivité du traitement du cas quasi-irrégulier peut être envisagée avec une stratégie régressive de manière à conserver un coût total en $O(n^2)$. Cette technique n'a d'intérêt que pour des sauts de taille importante, ce qui n'est pas le cas en général.

Chapitre 6

Application aux systèmes de Hankel et à l'algorithme d'Euclide

Nous rappelons d'abord quelques propriétés des matrices de Hankel dont nous nous sommes servis auparavant. Notre objectif est ensuite d'étendre les techniques du chapitre précédent à des systèmes plus généraux formés à partir de matrices de Hankel. La transition entre les polynômes orthogonaux et les systèmes de Hankel est décrite par exemple dans [46] ou [65], nous la détaillons dans le premier paragraphe. Le premier prolongement, étudié au paragraphe (6.5), consiste à choisir un second membre quelconque. Cependant la réalisation d'un algorithme régressif passe par l'étude préalable des systèmes de Hankel cycliques (6.4). On s'intéresse également en (6.7) aux familles de systèmes de Hankel dont les matrices sont basées sur la même fonctionnelle mais ayant un second membre présentant une structure de déplacement comme dans le cadre des polynômes orthogonaux. Dans le paragraphe suivant, on se penche sur le cas particulier des systèmes où la matrice de Hankel possède en plus une structure symétrique par rapport à l'anti-diagonale principale ce qui permet de supprimer un terme dans la relation de récurrence à trois termes. Le dernier paragraphe complète ce travail en s'intéressant à l'algorithme d'Euclide pour comprendre comment et pourquoi il est possible de l'envisager au moyen d'une stratégie progressive.

6.1 Inversion des matrices de Hankel

L'inverse d'une matrice de Hankel n'est en général pas une matrice de Hankel. Néanmoins, les inverses des matrices de Hankel constituent une classe de matrices caractéristiques: les matrices Bezoutiennes dont nous rappelons la définition.

Soient $p(x) = \sum_{i=0}^{i=n} p_i x^i$ et $q(t) = \sum_{i=0}^{i=n} q_i t^i$ deux polynômes en les variables x et t . Nous observons que le polynôme $p(x)q(t) - p(t)q(x)$ est divisible par $x - t$. En effet,

$$p(x)q(t) - p(t)q(x) = \sum_{i=0}^{i=n} \sum_{j=0}^{j=n} p_i q_j (x^i t^j - x^j t^i)$$

$$\begin{aligned}
&= \sum_{0 \leq i < j \leq n} (p_i q_j - p_j q_i) (x^i t^j - x^j t^i) \\
&= \sum_{0 \leq i < j \leq n} (p_i q_j - p_j q_i) \sum_{s=0}^{s=j-i-1} (x-t) x^{i+s} t^{j-s-1},
\end{aligned}$$

ce que l'on peut encore noter

$$p(x)q(t) - p(t)q(x) = (x-t) \sum_{i=1}^{i=n} \sum_{j=1}^{j=n} b_{ij} x^{i-1} t^{j-1} \quad (6.1)$$

On définit alors une forme bilinéaire symétrique sur \mathbb{C}^n par

$$B(p, q)(u, v) = \sum_{i=1}^{i=n} \sum_{j=1}^{j=n} b_{ij} u_i v_j \text{ pour tout } u, v \in \mathbb{C}^n.$$

Définition 4 La matrice de la forme bilinéaire précédente est appelée la matrice Bezoutienne associée aux polynômes p et q et on la note également $B(p, q)$.

Remarquons que l'on passe de la notation matricielle à la notation polynômiale par l'égalité ,

$$\frac{p(x)q(t) - p(t)q(x)}{x-t} = (1, x, \dots, x^n) B(p, q) (1, t, \dots, t^n)^T.$$

Il existe différentes démonstrations pour montrer que l'inverse d'une matrice de Hankel est une matrice Bezoutienne. Dans ([46]) la démonstration repose sur la décomposition des matrices de Hankel à l'aide de matrices de Vandermonde. Nous présentons ici une démonstration élémentaire qui découle uniquement de la relation de Christoffel-Darboux.

Considérons le cas d'une matrice de Hankel fortement régulière H_n associée à une fonctionnelle linéaire c . Si on note P la matrice dont les colonnes sont constituées par les coefficients des polynômes $\{P_i, i \leq n+1\}$ orthogonaux relativement à c on a

$$P^T H_n P = D_n$$

où D_n est une matrice diagonale. Partant de là, on obtient successivement

$$H_n = P^{T^{-1}} D_n P^{-1}$$

$$H_n^{-1} = P D_n^{-1} P^T$$

Passons en notations polynômiales.

$$\begin{aligned}
(1, x, \dots, x^n) H_n^{-1} (1, t, \dots, t^n)^T &= (1, x, \dots, x^n) P D_n^{-1} P^T (1, t, \dots, t^n)^T \\
&= (P_0(x), P_1(x), \dots, P_n(x)) D_n^{-1} (P_0(t), P_1(t), \dots, P_n(t))^T \\
&= \sum_{i=0}^{i=n} \frac{1}{h_i} P_i(x) P_i(t).
\end{aligned}$$

Appliquons la relation de Christoffel-Darboux (3.4).

$$\sum_{i=0}^{i=n} \frac{1}{h_i} P_i(x) P_i(t) = \frac{P_{n+1}(x) P_n(t) - P_{n+1}(t) P_n(x)}{x - t} \quad (6.2)$$

$$= (1, x, \dots, x^n) B(P_{n+1}, P_n) (1, t, \dots, t^n)^T. \quad (6.3)$$

Si nous envisageons le cas des matrices de Hankel seulement régulières, il faut utiliser une factorisation généralisée de la matrice de Hankel qui utilise par exemple les polynômes quasi-orthogonaux de Draux ([39], p.51) ainsi que la généralisation de la formule de Christoffel-Darboux ([39], p.81) au cas non régulier. La démonstration est alors identique. Les matrices Bezoutiennes satisfont une propriété très intéressante pour le calcul numérique: la formule de Gohberg-Semencul ([56]). Cette formule exprime une matrice Bezoutienne comme la somme de deux produits de matrices triangulaires de type Hankel et Toeplitz. Les démonstrations usuelles utilisent des calculs directs sur les coefficients ([67]). En fait, du point de vue des polynômes orthogonaux, la formule repose sur la notion de polynôme associé. Nous repartons de la définition d'une matrice Bezoutienne en polynôme de degré n de x et t ,

$$(1, x, \dots, x^n) B(P_{n+1}, P_n) (1, t, \dots, t^n)^T = \frac{P_{n+1}(x) P_n(t) - P_{n+1}(t) P_n(x)}{x - t} \quad (6.4)$$

dans laquelle nous faisons apparaître la définition des polynômes associés.

$$= P_{n+1}(x) \left(\frac{P_n(t) - P_n(x)}{x - t} \right) - P_n(x) \left(\frac{P_{n+1}(t) - P_{n+1}(x)}{x - t} \right). \quad (6.5)$$

Rappelons que le polynôme $Q_{n-1}(x)$ associé à $P_n(x)$ relativement à la fonctionnelle c est donné par,

$$Q_{n-1}(x) = c_t \left(\frac{P_n(t) - P_n(x)}{t - x} \right).$$

A l'aide des coefficients de $P_n(x) = \sum_{i=0}^{i=n} b_i x^i$ on peut exprimer matriciellement l'égalité précédente par,

$$Q_{n-1}(x) = (1, x, \dots, x^{n-1}) \begin{pmatrix} c_0 & c_1 & \dots & c_{n-2} & c_{n-1} \\ 0 & c_0 & \dots & c_{n-3} & c_{n-2} \\ \vdots & 0 & & \vdots & \vdots \\ 0 & \vdots & & c_0 & c_1 \\ 0 & 0 & \dots & 0 & c_0 \end{pmatrix} \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ \vdots \\ b_n \end{pmatrix}.$$

En échangeant les rôles du polynôme et de la fonctionnelle on obtient

$$Q_{n-1}(x) = (1, x, \dots, x^{n-1}) \begin{pmatrix} b_1 & b_2 & \dots & b_{n-1} & b_n \\ b_2 & b_3 & \dots & b_n & 0 \\ \vdots & \vdots & & \vdots & \vdots \\ b_{n-1} & b_n & 0 & \dots & 0 \\ b_n & 0 & \dots & 0 & 0 \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ \vdots \\ \vdots \\ c_{n-1} \end{pmatrix}.$$

Si on n'applique pas la fonctionnelle c_t , ou autrement dit, si la fonctionnelle a pour moments $c = 1, t, t^2, \dots, t^n$ alors on obtient l'expression matricielle de

$$Q_{n-1}(x, t) = \frac{P_n(t) - P_n(x)}{t - x} = (1, x, \dots, x^{n-1}, 0) \begin{pmatrix} b_1 & b_2 & \dots & b_n & 0 \\ b_2 & \dots & b_n & 0 & 0 \\ \vdots & & & & \vdots \\ b_n & 0 & \dots & 0 & 0 \\ 0 & 0 & \dots & 0 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ t \\ \vdots \\ \vdots \\ t^n \end{pmatrix}.$$

Considérons maintenant $P_{n+1}(x) = \sum_{i=0}^{n+1} a_i x^i$ et $Q_{n-1}(x, t)$ comme deux polynômes en la variable x et dont on veut représenter le produit modulo x^{n+1} sous forme matricielle. Puisque l'expression cherchée (6.4) est de degré n en x , on ne garde que les termes dont le degré en x est inférieur ou égal à n , il vient

$$P_{n+1}(x)Q_{n-1}(x, t) \bmod x^{n+1} = \\ -(1, x, \dots, x^{n-1}, x^n) \begin{pmatrix} a_0 & 0 & \dots & 0 & 0 \\ a_1 & a_0 & 0 & \dots & 0 \\ \vdots & \vdots & & \vdots & \\ a_{n-1} & a_{n-2} & \dots & a_0 & 0 \\ a_n & a_{n-1} & \dots & a_1 & a_0 \end{pmatrix} \begin{pmatrix} b_1 & b_2 & \dots & b_n & 0 \\ b_2 & \dots & b_n & 0 & 0 \\ \vdots & & & & \vdots \\ b_n & 0 & \dots & 0 & 0 \\ 0 & 0 & \dots & 0 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ t \\ \vdots \\ \vdots \\ t^n \end{pmatrix}.$$

On traite la deuxième partie de l'expression (6.5) comme la première pour obtenir finalement,

$$-P_n(x) \left(\frac{P_{n+1}(t) - P_{n+1}(x)}{x - t} \right) = \\ (1, x, \dots, x^{n-1}, x^n) \begin{pmatrix} b_0 & 0 & \dots & 0 & 0 \\ b_1 & b_0 & 0 & \dots & 0 \\ \vdots & \vdots & & \vdots & \\ b_{n-1} & b_{n-2} & \dots & b_0 & 0 \\ b_n & b_{n-1} & \dots & b_1 & b_0 \end{pmatrix} \begin{pmatrix} a_1 & a_2 & \dots & a_n & a_{n+1} \\ a_2 & a_3 & \dots & a_{n+1} & 0 \\ \vdots & \vdots & & \vdots & \\ a_n & a_{n+1} & \dots & 0 & 0 \\ a_{n+1} & 0 & \dots & 0 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ t \\ \vdots \\ \vdots \\ t^n \end{pmatrix}.$$

On peut donc conclure par une formule donnant l'inverse de H_n uniquement à l'aide des coefficients des polynômes orthogonaux P_{n+1} et P_n .

$$H_n^{-1} = B(P_{n+1}, P_n) \\ = \begin{pmatrix} b_0 & 0 & \dots & 0 & 0 \\ b_1 & b_0 & 0 & \dots & 0 \\ \vdots & \vdots & & \vdots & \\ b_{n-1} & b_{n-2} & \dots & b_0 & 0 \\ b_n & b_{n-1} & \dots & b_1 & b_0 \end{pmatrix} \begin{pmatrix} a_1 & a_2 & \dots & a_n & a_{n+1} \\ a_2 & a_3 & \dots & a_{n+1} & 0 \\ \vdots & \vdots & & \vdots & \\ a_n & a_{n+1} & \dots & 0 & 0 \\ a_{n+1} & 0 & \dots & 0 & 0 \end{pmatrix}$$

$$- \begin{pmatrix} a_0 & 0 & \cdots & 0 & 0 \\ a_1 & a_0 & 0 & \cdots & 0 \\ \vdots & \vdots & & & \vdots \\ a_{n-1} & a_{n-2} & \cdots & a_0 & 0 \\ a_n & a_{n-1} & \cdots & a_1 & a_0 \end{pmatrix} \begin{pmatrix} b_1 & b_2 & \cdots & b_n & 0 \\ b_2 & \cdots & b_n & 0 & 0 \\ \vdots & & & & \vdots \\ b_n & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 \end{pmatrix}.$$

Cette formule d'inversion est implicitement contenue dans l'algorithme de Trench ([102]), et la démonstration présentée simplifie grandement sa compréhension.

6.2 Méthodes de résolution des systèmes de Hankel

La résolution d'un système linéaire par l'inversion d'une matrice $n \times n$ nécessite $O(n^3)$ opérations élémentaires, en utilisant par exemple l'élimination gaussienne. Pour les systèmes de Toeplitz fortement réguliers les premières méthodes récurrentes en $O(n^2)$ remontent aux articles de Levinson [76] (1947) pour la résolution de systèmes de Toeplitz symétriques et Trench [102] (1965) pour l'inversion des matrices de Toeplitz fortement régulières. Ces algorithmes reposent sur la construction de récurrences obtenues par des méthodes de bordage qui exploitent essentiellement la structure très spéciale des matrices de Toeplitz. Comme on peut le constater dans les introductions de ([27],[33] et [85]) et leurs références, de nombreux auteurs ont apporté depuis leur contribution sur la question. Pour se donner une idée de la ferveur suscitée par le sujet, il faut prendre conscience des multiples domaines dans lesquels sont intervenus les systèmes de type Toeplitz ou Hankel. En analyse numérique, on peut citer par exemple les méthodes de Lanczos [63], les polynômes orthogonaux [39], les approximants de Padé [55], les méthodes d'extrapolation [26] ainsi que certaines équations aux dérivées partielles [79]. Pour le point de vue algébrique, on peut évoquer les codes correcteurs d'erreurs [73], les matrices Bezoutiennes [67] et l'algorithme d'Euclide [39]. En traitement du signal, on constate que le sujet apparaît en terme de filtrage et de prédiction linéaire dans les processus aussi variés que le traitement de la parole ou le traitement de signaux sismiques [30] et même la restauration d'images. La variété des sujets impliqués a permis d'obtenir une grande diversité dans les langages utilisés pour exprimer et résoudre les systèmes de Hankel. On trouve, outre l'aspect matriciel naturel ([46],[12]), des présentations utilisant par exemple les techniques associées aux polynômes orthogonaux [49] ou aux approximants de Padé [33]. Certains auteurs ont adopté un point de vue axé sur les fonctions génératrices [85]. Il n'y a pas une méthode privilégiée, mais chaque domaine apporte ses acquis et se trouve donc plus approprié pour aborder tel ou tel aspect de la question. Les méthodes ont progressé dans différentes directions. Tout d'abord, elles se sont étendues aux systèmes de Toeplitz non nécessairement fortement réguliers, avec par exemple les travaux de Berlekamp [4]. Récemment, dans ([12],[49]),[24]), les efforts se sont concentrés sur le contournement des sous-systèmes mal conditionnés rencontrés au cours du développement des récurrences ([33] et [7]). Toutes ces stratégies reposent sur un

bordage progressif. Une approche utilisant les approximants de Padé et une approche de division pour conquérir a également permis de réduire la complexité à $O(n \log^2(n))$ ([13] et [61]). De nouvelles stratégies ont également été étudiées, notamment par V. Pan [84] pour l'utilisation de la méthode de Newton, et G. Heinig par le biais des matrices de Cauchy et de Loewner [64]. Les dernières avancées proposées par K. Van Barel utilisent les matrices de Loewner et un pivotage des pôles de l'interpolation rationnelle sous-jacente ([74] et [103]). Enfin, du point de vue de la programmation, les efforts se portent actuellement sur la parallélisation des algorithmes précédents et de leur généralisation au cas vectoriel.

6.3 Liaison entre les systèmes de Hankel quelconques et les polynômes orthogonaux.

Nous exposons à présent une interprétation polynomiale de la formule de Gohberg qui permet de prolonger le lien avec l'identité de Bezout.

Dans le cas d'une fonctionnelle régulière c , les polynômes $\{P_i\}$ orthogonaux formels relativement à c et leurs associés $\{Q_i\}$ vérifient l'identité de Bezout. Nous allons montrer que l'identité de Bezout découle de l'inversion des matrices de Hankel et que l'on peut obtenir de nouvelles identités de type Bezout.

D'après (6.2) et pour n fixé, la matrice de Hankel H_n construite à partir de c admet une matrice inverse B_n bezoutienne: ce que l'on traduit par

$$B_n H_n = I_n.$$

D'après (6.9.6) et en écrivant les n équations reliant les n colonnes des deux membres de (6.9.6) pour $i = 0$ à $n - 1$

$$B_n(P_n, P_{n-1})(c_i, \dots, c_{i+n-1})^T = e_i^T.$$

En introduisant la variable x on obtient

$$(1, x, \dots, x^{n-1})B_n(P_n, P_{n-1})(c_i, \dots, c_{i+n-1})^T = x^i.$$

On interprète les coefficients c_i comme des moments de c évalués en la variable t pour obtenir

$$c_t^{(i)}((1, x, \dots, x^{n-1})B_n(P_n, P_{n-1})(1, t, \dots, t^{n-1})^T) = x^i.$$

La définition de B permet de faire intervenir directement P_n, P_{n-1} .

$$c_t^{(i)}\left(\frac{P_n(x)P_{n-1}(t) - P_n(t)P_{n-1}(x)}{x - t}\right) = x^i.$$

La fonctionnelle s'appliquant en t on peut extraire les polynômes en x de la façon suivante

$$c_t^{(i)}\left(\frac{P_n(x)P_{n-1}(t) - P_n(x)P_{n-1}(x) + P_n(x)P_{n-1}(x) - P_n(t)P_{n-1}(x)}{x - t}\right) = x^i.$$

$$P_n(x)c_t^{(i)}\left(\frac{P_{n-1}(t) - P_{n-1}(x)}{x - t}\right) + P_{n-1}(x)c_t^{(i)}\left(\frac{P_n(x) - P_n(t)}{x - t}\right) = x^i.$$

On reconnaît la définition de polynômes associés adjacents, c'est-à-dire

$$P_n(x)Q_{n-2}^{(i)}(x) + P_{n-1}(x)Q_{n-1}^{(i)}(x) = x^i. \quad (6.6)$$

Pour $i = 0$ nous retrouvons l'identité de Bezout.

On considère maintenant le système de Hankel suivant

$$H_n X = B \quad (6.7)$$

où B est un vecteur quelconque de \mathbb{C}^n . On suppose que H_n est régulière d'ordre n et associée à une fonctionnelle linéaire c .

6.4 Systèmes linéaires de type Hankel

On considère maintenant le système de Hankel suivant

$$H_n X = B \quad (6.8)$$

où B est un vecteur quelconque de \mathbb{C}^n . On suppose que H_n est régulière d'ordre n et associée à une fonctionnelle linéaire c .

Le chapitre 3 et les propriétés permettent d'entrevoir des méthodes régressives utilisant l'algorithme d'Euclide. Dans [20] la méthode de bordage inverse montre que les étapes régressives peuvent apporter de meilleurs résultats. Notre approche consiste à généraliser l'utilisation de la stratégie régressive en montrant qu'elle peut se substituer aux méthodes progressives. Il ne faudrait cependant pas voir l'algorithme régressif comme une simple récurrence renversée. La différence essentielle réside dans la manière d'utiliser la fonctionnelle linéaire. Pour le cas progressif, les moments sont introduits de façon récurrente, alors que pour l'algorithme régressif, la fonctionnelle n'entre en ligne de compte que pour la première itération. Ainsi, l'algorithme régressif apparaît comme une synthèse soustractive avec l'algorithme d'Euclide pour méthode de filtrage, alors que la récurrence à trois termes s'apparente à une synthèse additive. Nous suivons la même progression que celle décrite pour l'obtention de l'algorithme régressif pour le calcul des polynômes orthogonaux.

6.4.1 Systèmes linéaires de Hankel cycliques

Revenons au cas cyclique introduit dans la section (3.5). Nous rappelons que la matrice H_{2n} est associée à la fonctionnelle linéaire \tilde{c} dont les moments sont

$$(c_0, c_1, \dots, c_{2n-1}, c_0, c_1, \dots, c_{2n-1}).$$

On note $C^T = (c_{2n-1}, c_{2n-2}, \dots, c_1, c_0)$ et nous choisissons, pour le système (6.8), un second membre cyclique $B^T = (b_0, b_1, \dots, b_{n-1}, b_0, b_1, \dots, b_{n-1})$. En se référant à la section (2)

$$H_{2n}X = B_{2n}$$

devient

$$C * X = P_{2n}B_{2n}.$$

On obtient alors facilement que

$$X = F^{-1} \left(\frac{FP_{2n}B_{2n}}{FC} \right).$$

De la même façon, la méthode de bordage nous permet de calculer la solution du principal sous-système d'ordre $2n - 1$

$$H_{2n-1}Y_{2n-1} = B_{2n-1}$$

où $B_{2n-1}^T = (b_0, b_1, \dots, b_{n-1}, b_0, b_1, \dots, b_{n-2})$.

En bordant avec le vecteur $U^T = (c_{2n-1}, c_0, \dots, c_{2n-2})$ complété par un $y \in \mathbb{C}$ quelconque, on a

$$\begin{pmatrix} H_{2n-1} & U \\ U^T & c_{2n-2} \end{pmatrix} \begin{pmatrix} Y_{2n-1} \\ y \end{pmatrix} = \begin{pmatrix} H_{2n-1}Y_{2n-1} + yU \\ U^TY_{2n-1} + yc_{2n-2} \end{pmatrix}.$$

En d'autres termes

$$C * \begin{pmatrix} Y_{2n-1} \\ y \end{pmatrix} = \begin{pmatrix} U^TY_{2n-1} + yc_{2n-2} \\ P_{2n-1}(H_{2n-1}Y_{2n-1} + yU) \end{pmatrix}.$$

On modifie cependant le choix effectué dans la section 3.5.2, en posant maintenant $y = 0$. On obtient alors

$$FCF \begin{pmatrix} Y_{2n-1} \\ 0 \end{pmatrix} = F \begin{pmatrix} U^TY_{2n-1} \\ P_{2n-1}H_{2n-1}Y_{2n-1} \end{pmatrix}.$$

Lorsque FC n'a pas de composante nulle, on a

$$F \begin{pmatrix} Y_{2n-1} \\ 0 \end{pmatrix} = \frac{U^TY_{2n-1}Fe_1 + F \begin{pmatrix} 0 \\ P_{2n-1}H_{2n-1}Y_{2n-1} \end{pmatrix}}{FC}$$

où $e_1^T = (1, 0, \dots, 0)$ est le premier vecteur de la base canonique de \mathbb{R}^n . On peut dès à présent calculer implicitement Y_{2n-1} à l'aide de

$$\begin{pmatrix} Y_{2n-1} \\ 0 \end{pmatrix} = \alpha e_1 + V$$

avec

$$\alpha = \frac{U^TX_{2n-1}}{e_1^TFC}$$

et

$$V = \frac{F \begin{pmatrix} 0 \\ P_{2n-1} B_{2n-1} \end{pmatrix}}{FC}.$$

On commence par calculer V , puis on détermine α par la formule explicite

$$\alpha = \frac{-U^T V}{U^T e_1 - e_1^T FC}.$$

On doit encore montrer comment on obtient itérativement les solutions des sous-systèmes de taille inférieure.

6.5 De l'algorithme d'Euclide aux systèmes de Hankel

On considère un système régulier d'ordre $k+1$, associé à la fonctionnelle linéaire c de moments $(c_0, c_1, \dots, c_{2k})$

$$H_{k+1} Z_{k+1} = \begin{pmatrix} H_k & U_k \\ U_k^T & c_{2k} \end{pmatrix} \begin{pmatrix} Z'_k \\ z_{k+1} \end{pmatrix} = \begin{pmatrix} D_k \\ d_{k+1} \end{pmatrix} = D_{k+1} \quad (6.9)$$

et son principal sous-système d'ordre k , supposé également régulier

$$H_k Z_k = D_k. \quad (6.10)$$

Puisque nous avons besoin d'une relation entre Z_k et Z'_k , de (6.9) nous déduisons

$$H_k Z'_k + z_{k+1} U_k = D_k \quad (6.11)$$

et on pose

$$Z'_k = Z_k + R_k.$$

En tenant compte de (6.11) et de (6.10) nous obtenons

$$H_k R_k = -z_{k+1} U_k.$$

A l'exception du coefficient z_{k+1} , nous reconnaissons ici le système linéaire (1.4) qui définit P_k le polynôme unitaire de degré k orthogonal par rapport à c . Grâce à cette remarque, si on note \tilde{P}_k le vecteur constitué par les k premiers coefficients de P_k , on obtient

$$Z_k = Z'_k - z_{k+1} \tilde{P}_k. \quad (6.12)$$

Nous désignerons par $V(x)$ le polynôme associé au vecteur $V^T = (v_0, v_1, \dots, v_{k-1})$ de \mathbb{C}^k

$$V(x) = \sum_{i=0}^{k-1} v_i x^i.$$

L'idée principale est de comprendre la relation de récurrence (6.12) comme la division euclidienne du polynôme Z_{k+1} de degré k par le polynôme unitaire P_k de degré k , c'est-à-dire

$$Z_{k+1}(x) - z_{k+1}P_k(x) = Z_k(x). \quad (6.13)$$

Ceci nous permet de présenter un résultat plus général que le théorème (1) de Draux utilisé dans la section 3.2.

Théorème 3 *Pour un système de Hankel, deux sous-systèmes principaux réguliers d'ordres l_k et l_{k+1} , exclusivement séparés par d'éventuels sous-systèmes irréguliers, le polynôme $Z_{l_k}(x)$ de degré $l_k - 1$, dont les coefficients sont donnés par les composantes du vecteur Z_{l_k} solution du sous-système principal d'ordre l_k , est le reste de la division Euclidienne de $Z_{l_{k+1}}(x)$ de degré $l_{k+1} - 1$ par le polynôme $P_{l_k}(x)$ unitaire de degré l_k orthogonal par rapport à la fonctionnelle linéaire associée à la matrice de Hankel.*

preuve:

Etant donné les vecteurs $Z_{l_{k+1}}$ et P_{l_k} définis par l'énoncé, on peut calculer par la division Euclidienne les polynômes Q et R_{l_k} , uniques à un coefficient multiplicateur près, tels que

$$Z_{l_{k+1}}(x) - Q(x)P_{l_k}(x) = R_{l_k}(x)$$

avec $\deg(R_{l_k}) < l_k$ et nécessairement $\deg(Q) = l_{k+1} - 1 - l_k$.

Multiplier la matrice de Hankel $H_{l_{k+1}}$ d'ordre l_{k+1} du système de Hankel par le vecteur $Z_{l_{k+1}}$ correspond à évaluer les quantités

$$c(x^i Z_{l_{k+1}}(x)) \text{ pour } i = 0, \dots, l_{k+1} - 1.$$

En utilisant la linéarité de la fonctionnelle c associée à $H_{l_{k+1}}$, on a

$$c(x^i R_{l_k}(x)) = c(x^i Z_{l_{k+1}}(x)) - c(x^i Q(x)P_{l_k}(x)) \text{ pour } i = 0, \dots, l_{k+1} - 1. \quad (6.14)$$

Puisque les sous-systèmes d'ordre compris entre l_k et l_{k+1} sont singuliers et que $P_{l_{k+1}}$ est de degré l_{k+1} , en utilisant la régularité de P_{l_k} , on a

$$c(x^j P_{l_k}(x)) = 0 \text{ for } j = 0, \dots, l_{k+1} - 2.$$

On obtient ainsi

$$c(x^i Q(x)P_{l_k}(x)) = 0 \text{ pour } i + l_{k+1} - 1 - l_k \leq l_{k+1} - 2 \quad (6.15)$$

ce qui signifie plus précisément pour $i = 0, \dots, l_k - 1$.

Enfin, on introduit (6.15) dans (6.14). Par conséquent le vecteur R_{l_k} de dimension $l_k - 1$ associé au polynôme R_{l_k} a pour image par la restriction H_{l_k} de la matrice de Hankel de dimension l_k , un vecteur qui a les mêmes l_k premières composantes que l'image de $Z_{l_{k+1}}$ par $H_{l_{k+1}}$. Ceci nous permet de conclure que $R_{l_k} = Z_{l_k}$, à cause de la régularité de H_{l_k} .

6.6 Algorithme régressif pour les systèmes de Hankel

Algorithme 2

Etant donné les moments (c_i) , $i \in \{0, \dots, 2n - 1\}$ de la fonctionnelle linéaire c et le vecteur $B_n^T = (b_0, b_1, \dots, b_{n-1})$, cet algorithme calcule la solution du système $H_i X_i = B_i$ pour $i = 0, \dots, n$.

1. Poser $P_{2n}(x) = x^{2n} - 1$.
2. Poser $C^T = (c_{2n-1}, c_{2n-2}, \dots, c_0)$.
3. Calculer $V = F^{-1}(I/FC)$.
4. Poser $P_{2n-1}(x) = \sum_{i=0}^{2n-1} v_i x^i$.
5. Poser $D^T = (b_{n-1}, b_{n-2}, \dots, b_0, b_{n-1}, b_{n-2}, \dots, b_0)$.
6. Compute $W = F^{-1}(FD/FC)$.
7. Poser $Z_{2n-1}(x) = \sum_{i=0}^{2n-1} w_i x^i$.
8. Pour $i = 2n - 2, \dots, 1$, répéter
Calculer le reste $Z_i(x)$ de la division Euclidienne de $Z_{i+1}(x)$ par $P_{i+1}(x)$

Calculer le reste $P_i(x)$ de la division Euclidienne de $P_{i+2}(x)$ par $P_{i+1}(x)$
tant que $P_i(x) \neq 0$
9. Les P_i n'étant pas nécessairement unitaires, on obtient seulement des Z_i qui sont colinéaires aux solutions successives X_i . Si besoin est, la normalisation se fait ensuite.

6.6.1 Une relation à quatre termes

Dans le cas de forte régularité, on peut exhiber une relation à quatre termes satisfaite par les vecteurs solutions des sous-systèmes successifs. En partant des trois relations

$$\begin{cases} Z_{k+2}(x) - z_{k+2}P_{k+1}(x) = Z_{k+1}(x) \\ Z_{k+1}(x) - z_{k+1}P_k(x) = Z_k(x) \\ Z_k(x) - z_kP_{k-1}(x) = Z_{k-1}(x) \end{cases}$$

et de la relation à trois termes satisfaite par les polynômes orthogonaux unitaires (1.8), on trouve

$$\frac{1}{z_{k+2}}Z_{k+2}(x) - \left(\frac{x + \alpha_k}{z_{k+1}} + \frac{1}{z_{k+2}}\right)Z_{k+1}(x) + \left(\frac{x + \alpha_k}{z_{k+1}} - \frac{\beta_k}{z_k}\right)Z_k(x) = -\frac{\beta_k}{z_k}Z_{k-1}(x).$$

Le calcul de α_k et β_k à partir de Z_{k+2} , Z_{k+1} et Z_k ne nécessite pas le recours aux polynômes orthogonaux, puisqu'ils doivent uniquement être choisis de telle sorte que le membre de gauche de la récurrence précédente représente un polynôme de degré $k - 1$ exactement. Le principal avantage en comparaison de l'algorithme 2 est que l'on économise le coût d'un calcul en parallèle.

6.7 Application aux systèmes de Hankel en déplacement

Nous considérons à présent des familles de systèmes de Hankel qui généralisent la situation des polynômes orthogonaux. Nous entendons par systèmes de Hankel à déplacement le problème associé à une famille de systèmes de la forme

$$H_n X_n = B_n \tag{6.16}$$

où les H_n sont des matrices de Hankel de dimension $n + 1$ associées à la même fonctionnelle linéaire c de moments $(c_0, c_1, \dots, c_i, \dots)$ et où les seconds membres $B_n^T = (b_{n+1}, b_{n+2}, \dots, b_{2n+1})$ sont associés à une autre fonctionnelle linéaire b de moments $(b_0, b_1, \dots, b_i, \dots)$. Dans le cas particulier où $b = -c$, on retrouve le problème des polynômes orthogonaux. Remplacer n par $n + 1$ dans (6.16) définit ce que l'on nomme un *déplacement progressif*.

En utilisant la convolution et la FFT, on peut calculer encore une fois la solution associée au cas cyclique. Comme dans la section 3.5, on résout

$$H_{2n} X_{2n} = B_{2n}$$

avec $B_{2n}^T = (b_0, b_1, \dots, b_{2n})$. On obtient

$$X_{2n} = F^{-1} \begin{pmatrix} FP_{2n}B_{2n} \\ FP_{2n}C_{2n} \end{pmatrix}$$

avec $C_{2n}^T = (c_0, c_1, \dots, c_{2n})$.

En se restreignant au cas régulier, on utilise les procédés et les notations de la section 6.5 pour trouver une relation entre deux sous-systèmes successifs. On a d'une part

$$H_{k+1}Z_{k+1} = \begin{pmatrix} H_k & U_k \\ U_k^T & c_{2k+2} \end{pmatrix} \begin{pmatrix} Z'_k \\ z_{k+1} \end{pmatrix} = \begin{pmatrix} B'_k \\ b_{2k+3} \end{pmatrix} = B_{k+1}$$

où $B'_k{}^T = (b_{k+2}, b_{k+3}, \dots, b_{2k+2})$. D'autre part on a

$$H_k Z_k = B_k.$$

Du point de vue polynômial on trouve

$$c(x^i Z_{k+1}(x)) = b_{k+2+i} \text{ pour } i = 0, \dots, k+1$$

et

$$c(x^i Z_k(x)) = b_{k+1+i} \text{ pour } i = 0, \dots, k.$$

Ce que l'on peut encore écrire

$$c(x^i (xZ_k(x) - Z_{k+1}(x))) = 0 \text{ pour } i = 0, \dots, k-1.$$

Les polynômes $Q_k(x) = xZ_k(x) - Z_{k+1}(x)$ appartiennent à l'espace des polynômes de degré $k+1$ orthogonaux par rapport aux P_0, P_1, \dots, P_{k-1} relativement à c . Ce sous-espace est engendré par $P_k(x)$ et $P_{k+1}(x)$. Il existe donc des coefficients α_k and β_k tels que

$$xZ_k(x) - Z_{k+1}(x) = \alpha_k P_{k+1}(x) + \beta_k P_k(x). \quad (6.17)$$

Si on pose $x = 0$ dans (6.17), il ressort que

$$\alpha_k = \frac{-Z_{k+1}(0) - \beta_k P_k(0)}{P_{k+1}(0)}.$$

En prenant en compte cette relation dans (6.17), on obtient

$$\begin{aligned} Z_k(x) &= \frac{Z_{k+1}(x)P_{k+1}(0) - Z_{k+1}(0)P_{k+1}(x) + \beta_k(P_k(x)P_{k+1}(0) - P_k(0)P_{k+1}(x))}{xP_{k+1}(0)} \\ &= xQ_1(x) + \beta_k xQ_2(x) \end{aligned}$$

où $Q_1(x)$ et $Q_2(x)$ peuvent être aisément obtenus à partir de Z_{k+1} , P_{k+1} et P_k . Rappelons que

$$c(Z_k(x)) = b_{k+1}$$

de manière à trouver

$$\beta_k = \frac{b_{k+1}P_{k+1}(0) - c(Q_1(x))}{c(Q_2(x))}.$$

Théorème 4 Pour une famille de systèmes linéaires de Hankel fortement réguliers et à déplacement progressif, et deux sous-systèmes principaux de dimension $k + 1$ et $k + 2$, les polynômes Z_k de degré k , dont les coefficients sont donnés par les composantes de la solution de dimension $k + 1$, P_k et P_{k+1} les polynômes unitaires de degré k et $k + 1$ orthogonaux par rapport à la fonctionnelle associée à la matrice de Hankel du système, satisfont la récurrence régressive

$$xZ_k(x) - Z_{k+1}(x) = \alpha_k P_{k+1}(x) + \beta_k P_k(x)$$

avec

$$\beta_k = \frac{b_{k+1}P_{k+1}(0) - c(Q_1(x))}{c(Q_2(x))}$$

et

$$\alpha_k = \frac{-Z_{k+1}(0) - \beta_k P_k(0)}{P_{k+1}(0)}$$

où

$$Q_1(x) = \frac{Z_{k+1}(x)P_{k+1}(0) - Z_{k+1}(0)P_{k+1}(x)}{x}$$

et

$$Q_2(x) = \frac{P_k(x)P_{k+1}(0) - P_k(0)P_{k+1}(x)}{x}$$

L'algorithme basé sur ce théorème a la même structure que l'algorithme 2. Cette section tend à montrer que les mouvements dans la table des polynômes orthogonaux peuvent être généralisés par des mouvements dans les systèmes de Hankel.

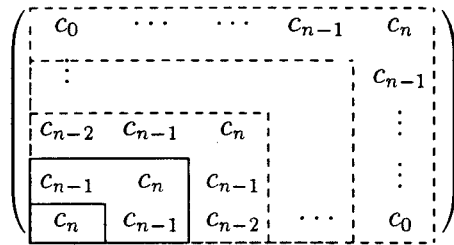
6.8 Application aux systèmes de Toeplitz symétriques

Les systèmes de Toeplitz symétriques ont une structure qui permet de simplifier la relation de récurrence à trois termes. Rappelons qu'un système de Toeplitz se ramène à un système de Hankel par simple permutation des lignes. Dans le cas d'un système de Toeplitz symétrique, le système de Hankel obtenu par ce biais dispose d'une symétrie supplémentaire qui se traduit par la symétrie de la fonctionnelle associée.

Définition 5 On dit que la fonctionnelle $c = c_0, c_1, \dots, c_{2n-1}, c_{2n-2}$ est symétrique par rapport au rang n si $c_{n-i} = c_{n+i}$ pour $i = 1, \dots, n$.

La relation de récurrence à trois termes ne prend pas en compte cette symétrie. Or nous avons observé en (1) que pour une fonctionnelle symétrique c , on a une propriété particulière qui permet de relier le polynôme $P_n^{(1)}$, unitaire de degré n , orthogonal adjacent par rapport à c et le polynôme \tilde{P}_n , normalisé en zéro de degré n , orthogonal par

rapport à c . Si on reconsidère alors la méthode de calcul des polynômes orthogonaux basée sur les récurrences croisées entre les familles de polynômes $P_n^{(1)}$ et \widetilde{P}_n , la nouvelle relation simplifie la récurrence de manière non négligeable. Les polynômes $P_n^{(1)}$ apparaissent systématiquement comme des polynômes \widetilde{P}_n renversés, ce qui en dispense le calcul. La stratégie qui permet de maintenir une symétrie permanente de la fonctionnelle est de faire croître la matrice de Hankel associée de manière homogène autour de la diagonale secondaire.



Les matrices de Hankel obtenues successivement en remontant la diagonale sont celles qui interviennent dans le calcul des polynômes, normalisés en $x = 0$, respectivement orthogonaux par rapport aux fonctionnelles notées

$$\begin{aligned}
 c^{(n-1)} &= \{c_{n-1}, c_n, \dots\} \\
 c^{(n-2)} &= \{c_{n-2}, c_{n-1}, c_n, c_{n-1}, \dots\} \\
 &\dots \dots \dots \\
 c^{(n-i)} &= \{c_{n-i}, c_{n-i+1}, \dots, c_{n-1}, c_n, c_{n-1}, c_{n-2}, \dots\} \\
 &\dots \dots \dots \\
 c^{(-1)} &= \{c_{-1}, c_0, c_1, \dots, c_{n-1}, c_n, c_{n-1}, \dots, c_1, c_0\}
 \end{aligned}$$

Tant que les fonctionnelles utilisées sont régulières, les polynômes orthogonaux $P_{k+1}(x)$, $P_k^{(1)}(x)$ et $P_k(x)$ définis par rapport à la même fonctionnelle symétrique

$$c^{(n-k+1)} = \{c_{n-k+1}, c_{n-k+2}, \dots, c_{n-1}, c_n, c_{n-1}, c_{n-2}, \dots, c_{n-k+1}, c_{n-k}\}$$

satisfont une relation de la forme

$$P_{k+1}(x) = xP_k^{(1)}(x) + \alpha P_k(x), \tag{6.18}$$

où $P_k^{(1)}(x) = x^k \widetilde{P}_k(\frac{1}{x})$ (6.9.6). Or $P_k(x)$ et $\widetilde{P}_k(x)$ sont identiques à un coefficient de normalisation près, donc on peut se ramener à la relation

$$P_{k+1}(x) = x(x^k \widetilde{P}_k(\frac{1}{x})) + \beta \widetilde{P}_k(x). \tag{6.19}$$

Remarquons que pour $P_{k+1}(x)$ la fonctionnelle n'est plus considérée comme symétrique, puisque le calcul utilise les moments

$$\{c_{n-k+1}, c_{n-k+2}, \dots, c_{n-1}, c_n, c_{n-1}, c_{n-2}, \dots, c_{n-k+1}, c_{n-k}, \}$$

et en respectant la normalisation en zéro on en déduit que

$$\widetilde{P}_{k+1}(x) = \gamma x(x^k \widetilde{P}_k(\frac{1}{x})) + \widetilde{P}_k(x). \quad (6.20)$$

Pour déterminer le coefficient γ on impose la condition d'orthogonalité

$$c(x^k \widetilde{P}_{k+1}(x)) = 0, \quad (6.21)$$

c'est-à-dire

$$\gamma = -\frac{x^k \widetilde{P}_k(x)}{x^{2k+1} \widetilde{P}_k(x)}. \quad (6.22)$$

L'obtention de γ nécessite le calcul de deux produits scalaires, et la récurrence n'utilise en fait qu'un seul vecteur. Pour parvenir au polynôme $\widetilde{P}_k(x)$, normalisé en zéro, orthogonal relativement à la fonctionnelle symétrique

$$c^{(-1)} = \{c_{-1}, c_0, c_1, \dots, c_{n-1}, c_n, c_{n-1}, \dots, c_1, c_0\}$$

(c_{-1} n'intervenant pas) on suit le schéma itératif suivant:

Etant donné le polynôme $P_i^{(1)}(x)$ de degré i orthogonal adjacent par rapport à $c^{(n-i)}$ et orthogonal relativement à la fonctionnelle symétrique

$$c^{(n-i+1)} = \{c_{n-i+1}, \dots, c_{n-1}, c_n, c_{n-1}, c_{n-2}, \dots, c_{n-i+1}\}$$

on déduit $\widetilde{P}_i(x)$, le polynôme normalisé en zéro, orthogonal relativement à la fonctionnelle

$$c^{(n-i)} = \{c_{n-i+1}, c_{n-i+1}, \dots, c_{n-1}, c_n, c_{n-1}, c_{n-2}, \dots, c_{n-i+1}\}$$

en renversant les coefficients de $P_i^{(1)}(x)$. Remarquons que $c^{(n-i+1)}$ est bien la fonctionnelle adjacente de $c^{(n-i)}$. La formule (6.9.6) nous donne le polynôme $P_{i+1}(x)$ de degré $i+1$ orthogonal relativement à la fonctionnelle symétrique $c^{(n-i)}$. En considérant ce dernier comme un polynôme orthogonal adjacent relativement à $c^{(n-i-1)}$ on peut renouveler le procédé.

On note $\widetilde{P}_k(x) = 1 + \sum_{i=1}^k a_i x^i$ et $\widetilde{P}_{k+1}(x) = 1 + \sum_{i=1}^{k+1} b_i x^i$. Le calcul de $\widetilde{P}_{k+1}(x)$ peut se représenter sous la forme:

$$\begin{pmatrix} 1 \\ a_1 \\ a_2 \\ \vdots \\ a_{k-1} \\ a_k \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ \gamma a_k \\ \gamma a_{k-1} \\ \vdots \\ \gamma a_2 \\ \gamma a_1 \\ \gamma \end{pmatrix} = \begin{pmatrix} 1 \\ b_1 \\ b_2 \\ \vdots \\ b_{k-1} \\ b_k \\ b_{k+1} \end{pmatrix} \quad (6.23)$$

Si k est pair, on a par symétrie pour $i = 1 \dots \frac{k}{2}$,

$$\begin{pmatrix} 1 & \gamma \\ \gamma & 1 \end{pmatrix} \begin{pmatrix} a_i \\ a_{n+1-i} \end{pmatrix} = \begin{pmatrix} b_i \\ b_{n+1-i} \end{pmatrix}. \quad (6.24)$$

Si k est impair, en désignant par s la partie entière de $\frac{k}{2}$, le système précédent est vérifié pour $i = 1 \dots s$, et

$$a_{s+1}(1 + \gamma) = b_{s+1}.$$

Ceci nous permet de remarquer que l'étape k nécessite $O(k)$ opérations arithmétiques, et que moyennant γ le calcul peut être réparti en parallèle sur k processeurs effectuant chacun 2 multiplications et 1 addition.

6.9 Application à l'algorithme d'Euclide progressif

6.9.1 Introduction

Ce paragraphe part du constat que le calcul des polynômes orthogonaux peut se faire au moyen de l'algorithme d'Euclide. On a naturellement tenté d'utiliser cette liaison en sens inverse afin d'obtenir les restes successifs dans l'algorithme d'Euclide par une méthode progressive basée sur les polynômes orthogonaux. Après avoir retracé un aperçu des différentes méthodes existantes pour calculer les restes successifs dans l'algorithme d'Euclide, nous exposons l'algorithme d'Euclide progressif et ses dérivés.

6.9.2 Généralités historiques

L'algorithme d'Euclide nous provient des éléments d'Euclide, un ensemble d'ouvrages mathématiques qui remonterait au 3ème siècle avant notre ère. Le premier objectif de l'algorithme est la comparaison des rapports de longueurs en les caractérisant par des suites de nombres entiers. Un rapport donné est grossièrement analysé par le plus grand nombre entier de fois qu'il faut utiliser le dénominateur pour recouvrir le numérateur sans le dépasser. Ce qui n'est pas recouvert, le reste, doit être analysé en rapport avec le dénominateur. Mais pour ce faire il faut inverser les rôles de sorte que le reste devient le dénominateur. Si le procédé itéré ne se termine pas on obtient la notion de longueurs incommensurables qui fut longtemps l'obstacle (les irrationnels) et la motivation de la construction du corps des réels (coupures de Dedekind). De nos jours, l'algorithme d'Euclide est usuellement présenté hors de ce contexte au travers de la structure axiomatique des anneaux euclidiens, ce qui permet d'en étendre l'application aux polynômes à une variable sur un corps en ce qui nous concerne. C'est cependant sous la forme des fractions continues que l'algorithme d'Euclide a connu son plus formidable essor. Les fractions continues ont d'abord été employées pour simplifier des rapports contenant des grands nombres, puis pour approcher des racines carrées. Initiée par Euler pour sommer des séries divergentes, la liaison avec les séries formelles a jeté un pont vers l'analyse. A travers l'étude analytique des fractions continues Stieljes posera les bases des polynômes orthogonaux.

Simultanément, son meilleur soutien C.Hermite emploie et développe des approximants rationnels qui prendront toute leur clarté avec les travaux de son élève H.Padé sur les approximants qui portent aujourd'hui son nom. Bien d'autres mathématiciens ont également apporté leurs contributions dans ces différents domaines en utilisant parfois les méthodes avant qu'elles ne soient formalisées. Pour les polynômes en une variable, quelle que soit la présentation choisie pour aborder l'algorithme, le cœur du travail consiste à construire une suite de restes polynomiaux (SRP). Nous présentons un panorama des techniques existantes pour calculer ces suites.

6.9.3 Les évolutions de l'algorithme d'Euclide

L'algorithme classique pour les polynômes reprend exactement la méthode utilisée pour les entiers, en construisant des polynômes de degrés de plus en plus petits dans l'idéal engendré par les deux polynômes de départ. Dans l'anneau Euclidien $K[X]$, la division euclidienne de $P_m(X) = \sum_{i=0}^m a_i X^i$ par $P_n(X) = \sum_{i=0}^n b_i X^i$ avec $m > n$ commence par éliminer le monôme de plus haut degré dans $P_m(X)$. On a

$$P_m - \frac{a_m}{b_n} P_n X^{m-n} = \frac{1}{b_n} (b_n P_m - a_m P_n X^{m-n}) = R_{m_1}$$

où R_{m_1} est de degré $m_1 \leq m-1$ et ses coefficients sont de la forme $\frac{r}{b_n}$ avec r dans l'anneau A des coefficients de P_m et P_n . En itérant le processus d'élimination, on constitue une suite de polynômes R_{m_i} de degrés m_i strictement décroissants et qui s'achève dès que $m_i < n$. Si on interprète cette étape en termes de polynômes orthogonaux, on est dans la situation inverse d'un saut dans la table des polynômes réguliers. Les polynômes intermédiaires R_{m_i} sont ceux utilisés par Draux pour compléter la table et qu'il nomme polynômes quasi-orthogonaux. Remarquons que si les coefficients de R_{m_i} sont de la forme $r/(b_n^i)$ avec $r \in A$ alors le même procédé prouve que les coefficients de $R_{m_{i+1}}$ sont de la forme $r/(b_n^{i+1})$ avec $r \in A$. Ceci permet de montrer par récurrence que les coefficients du reste de la division Euclidienne de P_m par P_n sont de la forme $r/(b_n^k)$ avec $r \in A$, où $k \leq m-n$. Cette remarque est à la base de la pseudo-division ([75], p. 369) qui maintient les calculs dans l'anneau des coefficients de départ. Il suffit de diviser $b_n^{m-n} P_m$ par P_n pour que tous les coefficients intermédiaires soient dans l'anneau des coefficients de départ. Le pseudo-reste obtenu est alors le reste de la division Euclidienne de P_m par P_n à un multiple b_n^i près. Au fond, le problème est le même que la recherche d'une normalisation appropriée dans le cadre des polynômes orthogonaux. Pour prolonger le parallèle on peut présenter une récurrence qui calcule des polynômes orthogonaux par rapport à une fonctionnelle c en conservant les coefficients dans l'anneau A des coefficients de la fonctionnelle c . On part d'une récurrence générale de la forme

$$P_{n+1} = (\alpha X + \beta) P_n + \gamma P_{n-1}. \quad (6.25)$$

Les deux dernières conditions d'orthogonalité nous donnent,

$$\alpha c(X^n P_n) + \gamma c(X^{n-1} P_{n-1}) = 0$$

et

$$\alpha c(X^{n+1}P_n) + \beta c(X^n P_n) + \gamma c(X^n P_{n-1}) = 0.$$

Pour rester dans l'anneau A des coefficients de c , à l'aide de la première équation on choisit $\alpha = -c(X^{n-1}P_{n-1})u$ et $\gamma = c(X^n P_n)u$ avec u à priori quelconque dans A . La deuxième condition permet de choisir alors u et β . En effet,

$$\beta c(X^n P_n) + u(c(X^n P_n)c(X^n P_{n-1}) - c(X^{n+1}P_n)c(X^{n-1}P_{n-1})) = 0$$

et on prend par exemple $u = -c(X^n P_n)$ et $\beta = c(X^n P_n)c(X^n P_{n-1}) - c(X^{n+1}P_n)c(X^{n-1}P_{n-1})$. D'autre part, si P_n et P_{n-1} ont des coefficients dans A alors les nombres $c(X^n P_n)$, $c(X^{n+1}P_n)$, $c(X^{n-1}P_{n-1})$ et $c(X^n P_{n-1})$ sont dans A comme produits scalaires et par suite les coefficients α , β , u et γ aussi, donc les coefficients de P_{n+1} obtenu par cette récurrence sont dans A . Les conséquences de cette remarque sont importantes, car cela nous conduit directement vers la conception d'algorithmes modulaires issus de ceux obtenus pour l'algorithme d'Euclide. La notion qui se dégage de tout cela est celle de *suite de restes polynômiaux* ([28]) ou SRP en abrégé.

Définition 6 *On qualifie de suite de restes polynômiaux dans $K[X]$ toute suite de polynômes P_i satisfaisant aux conditions suivantes*

$$\forall i \exists \beta_i \text{ tel que } \beta_i P_i = P_{i+1} - Q_i P_{i-1}$$

avec $\beta_i \in K$, $Q_i \in K[X]$ et $\deg(P_i) < \deg(P_{i+1})$.

Ceci englobe aussi bien les suites de restes obtenus par l'algorithme d'Euclide que les suites de polynômes orthogonaux et ce quelles que soient les normalisations adoptées au cours des calculs. Les premiers algorithmes utilisant cette notion proviennent de Collins ([36]). Si l'objectif est de rester dans $\mathbb{Z}[X]$, l'algorithme SRP qui découle immédiatement de la pseudo-division a malheureusement un effet catastrophique sur la croissance de la taille des coefficients. Par une première amélioration, l'algorithme SRP primitif, on cherche à vider les polynômes successifs de leur contenu, c'est-à-dire diviser par un diviseur commun des coefficients. Cependant cela nécessite de calculer des PGCD de coefficients et cela peut se révéler très lourd à mesure que les coefficients grossissent. Evidemment, on peut réaliser le même algorithme pour les polynômes orthogonaux. Si les coefficients de c sont dans \mathbb{Z} , on utilise la récurrence donnée en (6.25) et on divise P_{n+1} par son contenu à chaque étape. C'est pour modérer l'explosion des coefficients tout en limitant les calculs à effectuer pour diminuer le contenu des polynômes que Collins a introduit l'algorithme SRP réduit puis l'algorithme SRP sub-résultant. Dans ce dernier, les sub-résultants sont des déterminants de type Sylvester et permettent de majorer les coefficients par l'intermédiaire de l'inégalité de Hadamard. La construction des sub-résultants repose sur la décomposition de chaque terme de la suite dans l'idéal engendré par les deux polynômes initiaux. En fait, on peut exploiter cette décomposition pour se ramener au calcul de polynômes orthogonaux.

6.9.4 L'algorithme d'Euclide et les polynômes orthogonaux

Nous cherchons à construire une SRP $\{F_i\}$ dans $K[X]$ engendrée par F_1 et F_2 en calculant les F_i , de degrés n_i avec $n_{i+1} < n_i < n_2$, à partir de leurs décompositions dans l'idéal I engendré par F_1 et F_2 de degrés respectivement n_1 et n_2 avec $n_2 < n_1$. Soient A_i et B_i les polynômes tels que pour $i = 3, \dots, k$,

$$F_i = A_i F_1 + B_i F_2, \quad (6.26)$$

avec $\deg(A_i) = n_2 - n_{i-1}$, $\deg(B_i) = n_1 - n_{i-1}$. k est l'entier qui désigne la fin de la SRP, c'est-à-dire lorsque le pseudo-reste de la division de F_{k-1} par F_k est nul. En écrivant l'égalité polynomiale en la variable $1/X$ et en multipliant par $X^{n_1+n_2+n_{i-1}}$, on a

$$X^{n_1+n_2+n_{i-1}} F_i(1/X) = X^{n_2-n_{i-1}} A_i X^{n_1} F_1 + X^{n_1-n_{i-1}} B_i X^{n_2} F_2,$$

puis en utilisant les polynômes renversés,

$$X^{n_1+n_2-n_{i-1}-n_i} \tilde{F}_i = \tilde{A}_i \tilde{F}_1 + \tilde{B}_i \tilde{F}_2. \quad (6.27)$$

On peut alors interpréter le système (6.27) comme un problème d'approximation de Padé (1.10). En effet, la partie gauche de l'égalité précédente est en $O(X^{n_1+n_2-n_{i-1}-n_i})$ et la contrainte de minimisation des degrés pour obtenir cette approximation est vérifiée car $n_1 - n_i \geq n_1 - n_{i-1}$. Ainsi la fraction $-\tilde{B}_i/\tilde{A}_i$ est un approximant de Padé $[(n_1 - n_{i-1})/(n_2 - n_{i-1})]$ de la fraction rationnelle \tilde{F}_1/\tilde{F}_2 . Pour calculer les polynômes A_i et B_i on peut d'abord trouver la série qui décrit la fraction rationnelle \tilde{F}_1/\tilde{F}_2 . Ses coefficients forment la fonctionnelle par rapport à laquelle les dénominateurs renversés des approximants de Padé de \tilde{F}_1/\tilde{F}_2 sont orthogonaux (1.13). Le calcul des polynômes orthogonaux détermine les \tilde{A}_i dont les \tilde{B}_i se déduisent aisément soit par division, soit en résolvant un système de Hankel triangulaire. On revient alors aux polynômes F_i au moyen de la relation (6.26). J. R. Sendra et J. Llovet ([96]) ont appliqué une méthode semblable en passant uniquement par des systèmes de Hankel. Cependant, ils conduisent les calculs dans le but de trouver le PGCD et n'utilisent pas leur algorithme pour revenir aux polynômes F_i . Nous finalisons cette approche par deux remarques. Appliquée comme transition dans l'algorithme d'Euclide, la méthode peut servir à contourner les restes dont le calcul est mal conditionné, exactement comme dans le cas des polynômes orthogonaux. On pourrait être tenté de suivre une stratégie récurrente avec les polynômes associés aux F_i , ceux que l'on retrouve dans l'identité de Bezout en espérant éviter les problèmes provenant de certains F_i . Cependant, un petit coefficient dominant dans F_i entraîne une compensation par le coefficient dominant de l'associé de F_{i+1} .

6.9.5 Algorithme d'Euclide progressif

Etant donnés deux polynômes orthogonaux P_n et P_{n+1} , de degrés n et $n+1$ respectivement, on désire calculer la suite des restes de l'algorithme d'Euclide par une méthode progressive.

Algorithme d'Euclide progressif

L'algorithme calcule les polynômes orthogonaux par rapport à une fonctionnelle c pour laquelle P_n et P_{n+1} sont orthogonaux.

1. Déterminer l'ensemble des racines r_i du polynôme P_{n+1} .
2. Déterminer l'ensemble p_i des valeurs de P_n sur les racines r_i .
3. Calculer l'ensemble des valeurs q_i de Q_n sur les valeurs r_i en posant $q_i = 1/p_i$.
4. Calculer Q_n par interpolation sur les r_i .
5. Déterminer la fonctionnelle c qui associe Q_n à P_{n+1} .
6. Calculer les polynômes P_i orthogonaux relativement à c . Ces polynômes sont les restes cherchés.

Il faut noter cependant que l'interpolation ne peut s'effectuer que si P_n et P_{n+1} sont premiers entre eux, mais dans ce cas une simplification par le facteur commun permet de ramener le problème à la situation de polynômes premiers entre eux. L'interpolation accepte des racines multiples à condition d'utiliser une méthode de type Hermite. Pour déterminer la fonctionnelle on peut par exemple résoudre un système linéaire triangulaire.

6.9.6 Développements possibles

Les outils connus que nous avons employés tout au long de ce travail laissent peu de liberté de mouvement alors que celle-ci est nécessaire aux méthodes de pivotage. Par l'algorithme régressif exprimé en terme de choix de points d'interpolation nous avons tenté d'apporter un début de solution en réordonnant les points d'interpolation en amont d'un schéma de Newton. Avec les méthodes récentes de pivotage des points d'interpolation ([90] et [57]) on peut s'attendre à de nouveaux prolongements tels que ceux obtenus par K. VAN BAREL ([103],[74]) dans le cas de l'interpolation rationnelle.

Références

- [1] G. S. AMMAR, W. B. GRAGG, Implementation and use of the generalised Schur algorithm, dans *Computational and Combinatorial Methods in Systems Theory*, C. I. Byrnes and A. Lindquist, Elsevier, New York, 1986, 265-279.
- [2] G. A. BAKER, P. GRAVES-MORRIS, *Padé Approximants*, 2nd edition, Cambridge, New York, 1996.
- [3] E. H. BAREISS, Numerical solution of linear equations with Toeplitz and vector Toeplitz matrices, *Numer. Math.*, 13 (1969), 404-424.
- [4] E.R. BERLEKAMP, *Algebraic Coding Theory*, McGraw-Hill, New York, 1968.
- [5] B. BECKERMANN, On the numerical condition of polynomial bases: Estimates for the Condition Number of Vandermonde, Krylov and Hankel matrices, *Fachgebiet Mathematik vorgelegte Habilitationsschrift*, Universität Hannover, 1995.
- [6] B. BECKERMANN, Nonsymmetric difference operators and polynomials being orthogonal with respect to rectangular matrix valued measures.(à paraître)
- [7] B. BECKERMANN, The stable computation of formal orthogonal polynomials.(à paraître)
- [8] B. BECKERMANN, G. LABAHN, When are two numerical polynomials relatively prime?. à paraître, (1997).
- [9] W. BARTH, R. S. MARTIN, J. H. WILKINSON, Calculation of the Eigenvalues of a symmetric tridiagonal matrix by the method of bisection, *Numer. Math.*, 9 (1967), 386-393.
- [10] D. BINI, V. PAN, *Polynomial and Matrix Computation*, Birkhäuser, Boston, 1994.
- [11] R. R. BITMEAD, B. D. O. ANDERSON, Asymptotically fast solution of Toeplitz and related systems of equations, *Lin. Alg. Appl.*, 34(1980), 103-116.
- [12] A. W. BOJANCZYK, G. HEINIG, A multi-step algorithm for Hankel matrices. *J. of complexity*, 10 (1994), 142-164.

- [13] R. P. BRENT, F. G. GUSTAVSON, D. Y. Y. YUN, Fast solution of Toeplitz systems of equations and computation of Padé approximants, *J. Algo.*, 1(1980), 259-295.
- [14] C. BREZINSKI, Sur les polynômes associés à une famille de polynômes orthogonaux, *C. R. Acad Sc. Paris*, t. 284 (1977), 1041-1044.
- [15] C. BREZINSKI, *Padé-Type Approximation and General Orthogonal Polynomials*, Birkhäuser, Basel, 1980.
- [16] C. BREZINSKI, Outlines of Padé approximation, dans *Computational Aspects of Complex Analysis*, H.Werner et al. eds., Kluwer, Dordrecht, 1982, 1-50.
- [17] C. BREZINSKI, A direct proof of the Christoffel-Darboux identity and its equivalence to the recurrence relationship, *J. Comput. Appl. Math.*, 32 (1990), 17-25.
- [18] C. BREZINSKI, *Biorthogonality and its applications to Numerical Analysis*, Marcel Dekker, New York, 1991.
- [19] C. BREZINSKI, Generalizations of the Christoffel-Darboux identity for adjacent families of orthogonal polynomials, *Appl. Num. Math.*, 8 (1991), 193-199.
- [20] C. BREZINSKI, M. MORANDI.CECCHI, M.REDIVO-ZAGLIA, The reverse bordering method. *SIAM J.Matrix Anal. Appl.* ,15 (1994), 922-937.
- [21] C. BREZINSKI, C. MUSSCHOOT, Biorthogonal polynomials and the bordering method for linear system. Extrait de *Rendiconti del Seminario Matematico e Fisico di Milano* ,Vol. LXIV (1994).
- [22] C. BREZINSKI, M. REDIVO-ZAGLIA, A new presentation of orthogonal polynomials with application to their computation. *Numerical Algorithms*, 1 (1991), 207-222.
- [23] C. BREZINSKI, M. REDIVO-ZAGLIA, H. SADOK, Avoiding breakdown and near-breakdown in Lanczos type algorithms. *Numer.Algo.*, 1 (1991), 261-284.
- [24] C. BREZINSKI, M. REDIVO-ZAGLIA, H. SADOK, A breakdown-free Lanczos type algorithm for solving linear systems. *Numer. Math.*, 63 (1992), 29-38.
- [25] C. BREZINSKI, M. REDIVO-ZAGLIA, Breakdowns in the computation of orthogonal polynomials. Dans *Nonlinear Numerical Methods and Rational Approximation*, II, A. Cuyt ed., Kluwer, Dordrecht, (1994), 49-59.
- [26] C. BREZINSKI, M. REDIVO-ZAGLIA, *Extrapolation Methods. Theory and Practice*, North-Holland, Amsterdam, (1991).
- [27] J. R. BUNCH, Stability of methods for solving Toeplitz systems of equations, *SIAM J. Sci. Stat. Comp.*, 6 (1985), 349-364.

- [28] W. S. BROWN, On Euclid's algorithm and the computation of polynomial greatest common divisors, J.A.C.M., 18 (1971), 478-504.
- [29] A. BULTHEEL, M. VAN BAREL, Euclid, Padé and Lanczos, another golden braid, Report TW188, Department of Computer Science, K. U. Leuven, Avril(1993).
- [30] A. BULTHEEL, M. VAN BAREL , Linear prediction: mathematics and engineering. Bull. of the Belg. Math. Soc., 1(1994),1-58.
- [31] A. BULTHEEL, M. VAN BAREL, Formal orthogonal polynomials and Hankel/Toeplitz duality. Nmer. Algo., 10 (1995), 289-335.
- [32] A. BULTHEEL, Division algorithms for continued fractions and the Padé table, J. Comp. Appl. Math., 6 (1980), 259-266.
- [33] S. CABAY, RON MELESHKO, A weakly stable algorithm for Padé approximants and the inversion of Hankel matrices, SIAM J. Matrix Anal. Appl.,14 (1993), 735-765.
- [34] J.W. COOLEY, J.W. TUKEY, An algorithm for the machine calculation of complex Fourier series, Math. Comp. ,19 (1965), 297-301.
- [35] P. G. CIARLET, *Introduction à l'analyse numérique et d'optimisation* , Masson, Paris, 1982.
- [36] G. E. COLLINS, Subresultants and reduced polynomial remainder sequences, J. ACM, 14(1967),128-142.
- [37] J. DAVENPORT, Y. SIRET, E. TOURNIER, *Systèmes et Algorithmes de Manipulations Algébriques* , Masson, Paris, 1987.
- [38] P. DELSARTE, Y. GENIN, The split Levinson algorithm, IEEE Trans. Acoust. Speech Signal Process., ASSP-34,(1986), 470-478.
- [39] A. DRAUX, *Polynômes Orthogonaux Formels. Applications* , LNM 974, Springer-Verlag, Berlin, 1983.
- [40] A. DRAUX, P. VAN INGELANDT, *Polynômes Orthogonaux et approximants de Padé. Logiciels* , édition technip, Paris, 1987.
- [41] D. K. DUNAWAY, Calculation of zeros of a real polynomial through factorization using Euclid's algorithm, SIAM J. Numer. Anal., Vol.11, 6(1974) December.
- [42] J. DURBIN, The fitting of time-series models, Rev. Int. Stat. Inst., 28(1959), 229-249.
- [43] A. DUTT, M. GU, V. ROKHLIN, Fast algorithms for polynomial interpolation, integration and differentiation, SIAM J. Numer. Anal., Vol.33, 5(1996) 1689-1711.
- [44] J. L. CHABERT, *Histoire d'Algorithmes*, Belin, Paris (1993).

- [45] J. FAVARD, Sur les Polynômes de Tchebycheff, C. R. Acad. Sc., Paris 200 (1935), 2052-2055.
- [46] M. FIEDLER, *Special Matrices and their Applications in Numerical Mathematics*, Martinus Nijhoff Publishers, Dordrecht, 1986.
- [47] V. N. FADEEVA, *Computational Methods of Linear Algebra*, Dover, New York, 1959.
- [48] G.E. FORSYTHE, Generation and use of orthogonal polynomials for data-fitting with a digital computer, J. Soc. Indust. Appl. Math. , 5 (1957), 74-88.
- [49] R.W. FREUND, H. ZHA, A look-ahead algorithm for the solution of general Hankel systems, Numer. Math., 64 (1993), 295-321.
- [50] W. GAUTSCHI, Computational aspects of three-term récurrence relations. SIAM Rev. , 9 (1967), 24-82.
- [51] W. GAUTSCHI, Minimal solutions of three-term récurrence relations and orthogonal polynomials, Math.Comp. , 36 (1981), 547-554.
- [52] W. GAUTSCHI, On generating orthogonal polynomials, SIAM J. Sci. Stat. Comp, 3 (1982) 289-317.
- [53] W. GAUTSCHI, Orthogonal polynomials - Constructive theory and applications , J. of Comp. and Appl. Math., 12-13(1985) 61-76.
- [54] K. O. GEDDES, S. R. CZAPOR, G. LABAHN, *Algorithms for Computer Algebra*. Kluwer Acad. Pub., Dordrecht, (1992).
- [55] J. GILEWICZ, *Approximants de Padé*. Springer L. N. M. no. 667. Springer, Berlin, (1978).
- [56] I. C. GOHBERG, A. A. SEMENCUL, On the inversion of finite Toeplitz matrices and their continuous analogs, Mat. Issled., 2 (1972), 201-233.
- [57] I. C. GOHBERG, V. OLSHEVSKY, The fast generalized Parker-Traub algorithm for inversion of Vandermonde and related matrices, Mat. Issled., 2 (1972), 201-233.
- [58] G. H. GOLUB, C. F. VAN LOAN, *Matrix Computations*, Second Edition. The Johns Hopkins University Press, Baltimore, (1989).
- [59] G. H. GOLUB, M. H. GUTKNECHT, Modified moments for indefinite weight functions, Numer. Math., 57 (1990), 607-624.
- [60] W. B. GRAGG, The Padé table and its relation to certain algorithms of numerical analysis, SIAM Review, 14 (1972), 1-62.

- [61] W. B. GRAGG, F. G. GUSTAVSON, D. D. WARNER, D. Y. Y. YUN, On fast computation of superdiagonal Padé fractions, *Math. program. study*, 18 (1982), 39-42.
- [62] A. O. GUELFOND, The approximation of algebraic numbers by algebraic numbers and the theory of transcendental numbers. Extrait de *Number Theory and Analysis*, AMS, Rhode Island, Vol.2(1962).
- [63] M. H. GUTKNECHT, A completed theory of the unsymmetric Lanczos process and related algorithms, part 1, *SIAM J. Matrix Anal. Appl.*,13 (1992), 594-639.
- [64] G.HEINIG, Inversion of generalized Cauchy matrices and other classes of structured matrices, in *Proceedings of the IMA Workshop On Linear Algebra in Signal Processing*, Inst. for Math. and its Appl, University of Minnesota, Mineapolis (1992).
- [65] G.HEINIG, K.ROST, *Algebraic Methods for Toeplitz-like Matrices and Operators*, Birkhäuser, Basel, (1984).
- [66] G.HEINIG, P.JANKOWSKI, Parallel and superfast algorithms for Hankel systems of equations, *Numer. Math.*, 58 (1990), 109-127.
- [67] H. HELMKE, P. A. FUHRMANN, Bezoutians, *Lin. Algebra and its Appl.*, 122/123/124 (1989), 1039-1097.
- [68] HIGHAM, Fast solution of Vandermonde-like systems involving orthogonal polynomials, *IMA J. Numer. Anal.*, 8 (1988), 473-486.
- [69] H. HOCHSTADT, *Les fonctions de la Physique Mathématique*, Masson et cie, Paris, (?).
- [70] R. A. HORN, C. R. JOHNSON, *Matrix Analysis*, Cambridge University Press, Cambridge, 1993.
- [71] I. IOHVIDOV, *Hankel and Toeplitz Matrices and Forms - Algebraic Theory*, Birkhauser, Boston, 1982.
- [72] J.JÁJÁ, *An Introduction to Parallel Algorithms*, Addison-Wesley, Reading, Mass., 1992.
- [73] E. JONCKHEERE, C. MA, A simple Hankel interpretation of the Berlekamp Massey algorithm, *Lin. Algeb. Appl.*, 125(1989) 65-76.
- [74] P. KRAVANJA, K. VAN BAREL, A fast Hankel solver based on an inversion formula for Loewner matrices, Submitted, July 1996.
- [75] D.E. KNUTH, *The Art of Computer Programming, Vol 1: Fundamental Algorithms*, Addison-Wesley, Reading, Mass., 1973.

- [76] N.LEVINSON, The Wiener RMS error in filter design and prediction, J. Math. Phys., 25 (1947), 261-278.
- [77] G.LABAHN, D.K. CHOI, S. CABAY, The inverses of block Hankel and block Toeplitz matrices, SIAM J. Comput., 19 (1990), 98-123.
- [78] L. LORENTZEN, *Continued Fractions with Applications*, North-Holland, Amsterdam, 1992.
- [79] M. A. MALCOLM, J. PALMER, A fast method for solving a class of tridiagonal systems, Comm. ACM, 17 (1974), 14-17.
- [80] J. MILKOSKO, An algorithm for calculating continued fractions, J. Comp. Appl. Math., 3 (1977), 273-275.
- [81] G. MERZ, Fast Fourier transform algorithms with applications, dans *Computational Aspects of Complex Analysis*, H.Werner et al. eds., Kleuver, Dordrecht, 1982, 249-278.
- [82] J. G. NAGY, Iteratives techniques for the solution of Toeplitz systems, SIAM News, August (1995), 8-9.
- [83] J. G. NAGY, Applications of Toeplitz systems, SIAM News, October (1995), 10-12.
- [84] V. PAN, Parallel solution of Toeplitzlike linear systems , J. of complexity, 8 (1992), 1-21.
- [85] D. PAL, T. KAILATH, Fast triangular factorization and inversion of Hankel and related matrices with arbitrary rank profile, SIAM J. Matrix Anal. Appl., 15 (1994), 451-478.
- [86] O. PAPINI, J. WOLFMANN, *Algèbre Discrète et Codes Correcteurs*, Springer, Paris, 1994.
- [87] D. PERRIN, *Cours d'Algèbre*, Collection de l'Ecole Normale Supérieure de Jeunes Filles N18, ENS Paris, 1985.
- [88] W. PRESS, B. FLANNERY, S. TEUKOLSKY AND W. VETTERLING, *Numerical Recipes*, Cambridge University Press, Cambridge, 1988.
- [89] A. POLI, L. HUGUET, *Codes correcteurs*, Masson, Paris, 1989.
- [90] L. REICHEL, Newton interpolation at Leja points, BIT,30(1990), 23-41.
- [91] O. RENAULT, A new algorithm to compute orthogonal polynomials, J. Comput. Appl. Math., 75 (1996), 231-248.
- [92] J. R. RICE, Experiments on Gram-Schmidt orthogonalization, Math. Comp., 20 (1966), 325-328.

- [93] J. RISSANEN, Algorithms for triangular decomposition of block Hankel and Toeplitz matrices with applications of factoring positive polynomials, *Math. Comp.*, 27 (1973), 147-154.
- [94] H. RUTISHAUSER, Der Quotienten-Differenzen-Algorithms, *Z. Angew. Math. Physik*, 5 (1954), 233-251.
- [95] R. A. SACK, A. F. DONOVAN, An algorithm for Gaussian quadrature given modified moments, *Numer. Math.*, 18 (1972) 465-478.
- [96] J. R. SENDRA, J. LLOVET, An extended polynomial GCD algorithm using Hankel matrices, *J. Symb. Comp.*, 13 (1992), 25-39.
- [97] J. A. SHOHAT, Sur les polynômes orthogonaux généralisés, *C. R. Acad. Sci. Paris*, 207 (1938), 556-558.
- [98] M. SIBONY, J. C. MARDON, *Approximations et équations différentielles*, Hermann, Paris, 1984.
- [99] G. STRANG, A proposal for Toeplitz matrix calculations, *Stud. Appl. Math.*, 74 (1986), 171-176.
- [100] G. SZEGÖ, *Orthogonal Polynomials*, 4th ed., Amer. Math. Soc., Providence, RI, 1975.
- [101] R. THÉODOR P. LASCAUX, *Analyse Numérique Matricielle Appliquée à l'Art de l'Ingénieur*, Tome 1 et 2, Masson ed., Paris, 1986.
- [102] W.F.TRENCH, An Algorithm for inversion of finite Hankel matrices, *J. Soc. Indust. Appl. Math.*, 13 (1965), 1102-1107.
- [103] K. VAN BAREL, P. KRAVANJA, A A stabilized superfast solver for indefinite Hankel systems, Submitted, September 1997.
- [104] J. VAN ISEGHEM, Récurrence relations in the table of vector orthogonal polynomials. Dans *Nonlinear Numerical Analysis and Rational Approximation, II*, A. Cuyt ed., Kluwer, Dordrecht, 1994, 61-69.
- [105] H. K. WIMMER, On the History of the Bezoutian and the Resultant Matrix, *Lin. Algebra. and its Appl.*, 128 (1990), 27-34.
- [106] J.WIMP, *Computation with Récurrence Relations*, Pitman, Boston, 1984.
- [107] S. WINOGRAD, On computing the discrete Fourier Transform, *Math. Comput.*, 32 (1978), 175-199.

