LABORATOIRE D'ANALYSE
NUMÉRIQUE ET D'OPTIMISATION

N° d'ordre : 2360

THÈSE

présentée et soutenue publiquement à

L'UNIVERSITÉ DES SCIENCES ET TECHNOLOGIES DE LILLE

pour obtenir le grade de

DOCTEUR ES SCIENCES MATHÉMATIQUES

par

Christophe MUSSCHOOT

le 6 Janvier 1999.

Polynômes biorthogonaux : interprétation matricielle et résolution de systèmes linéaires à seconds membres multiples.

Directeur de thèse : Professeur Claude Bresinski.

Président :	C. BREZINSKI (Université de Lille I)
Rapporteurs :	A. DRAUX (INSA Rouen)
	M. REDIVO ZAGLIA (Université de Padoue – Italie)
Membres du Jury :	G. MEURANT (CEA)
	H. SADOK (Université du Littoral)
	J. VAN ISEGHEM (Université de Lille I)

à Éliane,
Roland,
Isabelle.

Je remercie Madame Simoncini, de l'université de Bologne, de m'avoir conseillé pour l'amélioration de la quatrième partie de ce travail. Ses remarques judicieuses m'ont été d'une aide précieuse. Ses travaux m'ont de plus considérablement éclairé sur le sujet.

Je remercie Madame Redivo Zaglia, professeur à l'université de Padoue, pour l'attention toute particulière qu'elle a pu accorder à la lecture de cette thèse en français, même si c'est une langue qu'elle maîtrise parfaitement. Ses remarques et commentaires m'ont été d'une aide précieuse.

Je remercie également Monsieur Draux, professeur à l'INSA de Rouen, pour les corrections nécessaires qu'il m'a conseillé d'effectuer dans cette thèse.

Je remercie tous mes collègues du laboratoire d'Analyse Numérique et d'Optimisation pour l'accueil qu'ils m'ont fait au sein de leur équipe et notamment madame Van Iseghem pour les démarches qu'elle a effectuées lors de la répartition des postes d'ATER.

Je remercie tout particulièrement Monsieur Brezinski, directeur du laboratoire ANO de l'université de sciences et technologies de Lille, de m'avoir considérablement aidé pour mon travail de recherche de documents, pour son infinie patience ainsi que pour tous ses conseils concernant l'élaboration, tant sur le fond que sur la forme, de cette thèse. Sans son soutien précieux, ce travail n'aurait peut-être jamais abouti.

Je remercie enfin Monsieur Sadok, professeur à l'université du Littoral, et Monsieur Meurant, docteur d'État au CEA de Bruyères le Chatel, d'avoir accepté de faire partie de mon jury.

Résumé.

Cette thèse comporte, dans son premier aspect, une généralisation des résultats matriciels connus sur les polynômes orthogonaux et polynômes orthogonaux de dimension $d > 1$ aux polynômes biorthogonaux.

Notamment, les notions de noyau reproduisant, d'identité de type Christoffel-Darboux seront abordées. Des propriétés matricielles des zéros des polynômes biorthogonaux seront également mises en évidence. Ainsi, certains résultats connus sur les polynômes orthogonaux et les polynômes vectoriellement orthogonaux restent valables tandis que d'autres ne le sont plus en général.

Une relation sera de plus établie entre la méthode de bordage et le calcul des polynômes biorthogonaux. Les matrices de Hankel et de Toeplitz seront traitées dans ce cadre.

Le second aspect de cette thèse est relatif à la résolution de systèmes linéaires à seconds membres multiples. Ainsi, une méthode basée sur une modification de la méthode de Lanczos et utilisant les polynômes biorthogonaux sera mise au point.

Un résultat notamment sur des algorithmes sans utilisation de la transposée pour la résolution de systèmes linéaires de ce type sera ainsi établi. Pour cela, des sous espaces de Krylov seront tout naturellement considérés.

Mots clés.

Polynômes orthogonaux et vectoriellement orthogonaux, polynômes biorthogonaux, noyau reproduisant, identité de type Christoffel-Darboux, zéros des polynômes biorthogonaux, matrices de Hankel, matrices de Toeplitz, systèmes à seconds membres multiples, sous espaces de Krylov, méthode de Lanczos, algorithme sans utilisation de la transposée.

Abstract.

This thesis gives, in its first aspect, a generalization of known results about the matrix interpretation of orthogonal and vector orthogonal polynomials to biorthogonal polynomials.

Such notions as the reproducing kernel and Christoffel-Darboux type formulae will be used. Some matrix properties on the zeros of biorthogonal polynomials will be shown. Thus, some known results on orthogonal polynomials and orthogonal polynomials of dimension $d > 1$ are still valid for biorthogonal polynomials while some are not.

Then, a relation will be established between the bordering method and the computation of biorthogonal polynomials. In particular, Hankel and Toeplitz matrices will be treated.

The second aspect of this work is the solution of linear systems with multiple right-hand sides. Thus, a method based on a modification of the Lanczos method and using biorthogonal polynomials will be given.

A result on transpose–free algorithms for solving linear systems with multiple right-hand sides is discussed. Krylov subspaces will be considered.

Keywords.

Orthogonal polynomials and vector orthogonal polynomials , biorthogonal polynomials, reproducing kernel, Christoffel-Darboux type formula, zeros of biorthogonal polynomials, Hankel matrices, Toeplitz matrices, multiple right hand sides linear systems, Krylov subspaces, Lanczos' method, transpose–free algorithm.

Table des matières

Introduction générale

En analyse numérique, les matrices de Hankel jouent un rôle tout particulier et sont fréquemment rencontrées (problèmes de moindres carrés, accélération de la convergence, approximants de Padé, théorie des systèmes ...). C'est pourquoi une littérature abondante sur ce sujet peut être trouvée quant à l'inversion de telles matrices (voir Fuhrmann [39], Gemignani [40], Lascoux [49] et Trench [70] entre autres). Les matrices de Toeplitz qui leur sont étroitement liées ont été également largement étudiées (voir par exemple les travaux de Trench [69] mais également de Cline et al. [29], de Dickinson [31] et de Kailath et al. [46]). Ces deux types de matrices ont été associés respectivement aux polynômes orthogonaux formels et orthogonaux formels de dimension -1. Des systèmes où des matrices formées à partir des matrices de Hankel et de Toeplitz (système dont la matrice est une somme de ces deux types) ont même été étudiés par R.H. Chan et al. [25].

Certaines propriétés, notamment les relations de récurrence, des polynômes orthogonaux ont été pleinement étudiées par Draux [32], tandis qu'une interprétation matricielle a été mise en évidence par Gragg [41] et par Brezinski [4].

Plus récemment, une généralisation des polynômes orthogonaux a été développée par Van Iseghem et la notion de polynômes orthogonaux de dimension $d > 1$ est apparue [73, 74, 75] et a également été abordée par Maroni [52] . La notion de polynômes orthogonaux vectoriels a même été étudiée (voir Draux et al. [33] et Le Ferrand [35]). L'orthogonalité matricielle a également été abordée par Van Iseghem et al. [77].

On peut encore étendre la notion de polynômes orthogonaux de dimension $d > 1$. On trouve alors les polynômes biorthogonaux introduits par Brezinski et étudiés quant à leur application à l'analyse numérique dans [8] et plus précisément pour la résolution de systèmes linéaires [9]. À ce jour, aucune interprétation matricielle n'a été fournie pour les polynômes biorthogonaux, même si des résultats partiels ont été obtenus par Bultheel et al. [24]. C'est ce que nous nous proposons de développer dans le premier aspect de cette thèse, ce qui donnera donc des résultats dans le cadre beaucoup plus général et unificateur des polynômes biorthogonaux.

C'est ainsi que des notions telles qu'identité de type Christoffel-Darboux, noyau reproduisant, valeurs propres, vecteurs propres, polynômes biorthogonaux à droite et à gauche seront abordées. Une relation sera également mise en évidence entre polynômes biorthogonaux et méthode de bordage.

Les polynômes orthogonaux, qui sont des polynômes biorthogonaux particuliers sont utilisés pour la résolution de systèmes linéaires. C'est le cas par exemple des méthodes de type Lanczos [48].

Récemment, beaucoup d'intérêt a été porté à la résolution des systèmes linéaires à seconds membres multiples. Ils ont été principalement basés sur des versions "bloc" de méthodes existantes pour la résolution de systèmes linéaires à second membre unique. C'est notamment le cas du *Block CG* (Block Conjugate Gradient) [50] pour les matrices symétriques mais également du *Block Bi-CG* (Block Bi-Conjugate Gradient) [50], des méthodes *Block QMR* (Block Quasi Minimal Residual) [3, 61], du *Block GMRES* (Block Generalized Minimal RESidual) [62, 64, 65] et également du *Block GCR* (Generalized Conjugate Residual) issu du GCR de Saad [58] pour les matrices quelconques.

Plus récemment encore, de nouvelles méthodes originales de résolution de systèmes linéaires à seconds membres multiples sont apparues, qui permettent une "communication" entre les différents systèmes à résoudre. C'est notamment le cas de celles établies par Simoncini et al. [63] et par T. F. Chan et al. [26] qui utilisent les résultats obtenus pour la résolution de certains des systèmes considérés pour la résolution des autres.

Afin d'éviter, pour des raisons que nous détaillerons ultérieurement, les résolutions par bloc, nous verrons que la méthode de Lanczos peut être modifiée pour la résolution de tels problèmes. Les polynômes biorthogonaux seront alors tout particulièrement considérés ainsi que les méthodes de type Lanczos. Il en découlera ainsi de nouveaux algorithmes pour la résolution de systèmes linéaires à seconds membres multiples.

Voyons maintenant comment s'articule cette thèse. Le plan général est défini comme suit.

Dans la **première partie**, nous rappelons les principaux résultats matriciels connus sur les polynômes orthogonaux. Pour cela, nous ferons référence aux travaux de Draux [32] et de Brezinski [4]. Ensuite, nous rappellerons également les résultats majeurs connus sur les polynômes orthogonaux de dimension $d > 1$ en faisant essentiellement référence aux travaux de Van Iseghem [73, 74, 75]. Certaines définitions et notations nécessaires pour la suite seront également abordées.

Ces rappels effectués, il sera possible de mettre en parallèle les résultats que nous obtiendrons sur les polynômes biorthogonaux dans la **deuxième partie**. Les notions de polynômes biorthogonaux à droite et polynômes biorthogonaux à gauche seront alors considérées. Certains résultats sur les polynômes biorthogonaux seront ainsi des généralisations de ceux sur les polynômes orthogonaux et orthogonaux de dimension $d > 1$ alors que d'autres ne pourront être étendus. Il sera alors intéressant de voir de quelle manière la généralisation s'opère, quand cela est le cas. D'autre part, nous verrons la relation qui existe entre le calcul des polynômes biorthogonaux et la mise en œuvre de la méthode de bordage.

Nous aborderons, à partir de la **troisième partie** le second aspect de cette thèse. Ainsi, les principales méthodes de résolution de systèmes linéaires à seconds membres multiples (*Block CG* et *Block Bi-CG* [50], *Block GMRES* [62, 64] et autres méthodes du type *Block QMR* [61] pour les versions "bloc" mais également les récentes méthodes de Simoncini et al. [63] et T. F. Chan et al. [26, 27]) seront considérées afin de pouvoir en apprécier les principales caractéristiques.

Enfin, dans la **quatrième partie**, après avoir rappelé la méthode de Lanczos [48], nous définirons une méthode de résolution de systèmes linéaires à seconds membres multiples à partir de la méthode de Lanczos et basée sur les polynômes biorthogonaux et les sous-espaces de Krylov. Nous verrons notamment que certains des algorithmes issus de cette méthode sont des algorithmes qui n'utilisent ni la transposée ni les puissances itérées de la matrice de départ, afin d'optimiser la stabilité de ces algorithmes. Le lien sera fait entre cette méthode et les méthodes de type Lanczos pour la résolution des systèmes à second membre unique comme le CGS (Conjugate Gradient Square) de Sonneveld [68] et le BiCG-Stab (Bi-Conjugate Gradient Stabilized) de Van Der Vorst [71]. Une comparaison numérique sur certains exemples des algorithmes définis dans cette partie avec la méthode du *Block Bi-CG* sera de plus effectuée.

Première partie
Interprétation matricielle des polynômes orthogonaux formels et polynômes vectoriellement orthogonaux

Introduction

Le but de cette partie est de rappeler les principaux résultats connus sur l'interprétation matricielle des polynômes orthogonaux et des polynômes orthogonaux de dimension $d > 1$, essentiellement mis en évidence dans [4] et [41] pour les premiers et dans [73, 74, 75] pour les seconds.

Il n'est pas question ici de décrire *in extenso* toutes les propriétés connues sur les polynômes orthogonaux et orthogonaux de dimension $d > 1$. Pour cela, il sera utile de consulter [4], [32], [41], [77] ou encore [76]. Nous n'aborderons que l'aspect qui nous intéresse par la suite, c'est-à-dire l'aspect matriciel, même si quelques notions supplémentaires seront nécessaires (notamment certaines relations de récurrence).

Ainsi, la **première section** rappellera les relations matricielles connues sur les polynômes orthogonaux formels. Nous considérerons tout d'abord quelques définitions avant de donner les principaux résultats matriciels concernant ces derniers, notamment ceux figurant dans [4] et [41].

Tandis que la **seconde section** énoncera les principaux résultats matriciels connus sur les polynômes orthogonaux de dimension $d > 1$. Nous serons donc à même de déterminer les résultats "perdus" lorsque l'on passe des polynômes orthogonaux aux polynômes vectoriellement orthogonaux.

1 Polynômes orthogonaux formels

Avant d'aborder les relations matricielles connues sur les polynômes orthogonaux formels, il est nécessaire de rappeler quelques définitions relatives à ces derniers, ce que nous ferons dans la **première sous-section.**

La **seconde sous-section** présentera, quant à elle, les résultats matriciels majeurs connus sur ces polynômes.

1.1 Définitions

L'interprétation matricielle des polynômes orthogonaux formels (qui pourront être désignés par abus de language "polynômes orthogonaux", même si ces derniers sont relatifs à des intégrales définies par rapport à une fonction poids positive, bornée sur un intervalle donné) et biorthogonaux contient tellement de notions diverses qu'il est nécessaire d'introduire quelques définitions afin d'en mesurer pleinement la portée.

Nous aborderons ici aussi bien des définitions essentielles que certaines relations majeures concernant les polynômes orthogonaux.

Tout d'abord, $\mathcal{P}$ désignera l'espace vectoriel des polynômes à coefficients complexes à une indéterminée ($\mathcal{P} = \mathbb{C}[X]$) et $\mathcal{P}_k$ le sous-espace vectoriel de $\mathcal{P}$ formé des polynômes de degré $\leqslant k$.

Définition 1.1 – *Fonctionnelle linéaire*

On appelle fonctionnelle linéaire une forme linéaire définie en général sur un espace fonctionnel quelconque et en particulier sur $\mathcal{P}$. Ainsi, la fonctionnelle linéaire c sera définie sur $\mathcal{P}$ par

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

où $c_0, c_1, \ldots$ sont des complexes et sont appelés les moments de c.

La notion de polynômes orthogonaux étant étroitement liée à celle de fonctionnelle linéaire, la définition d'une suite ou famille de polynômes orthogonaux peut désormais être énoncée.

Définition 1.2 – *Polynômes orthogonaux formels*

Soit $\{P_k\}_{k\geqslant 0}$ une suite d'éléments de $\mathcal{P}_k$. Alors, si les polynômes P_k, $\forall k$, vérifient

$$c(x^i P_k) = 0 \text{ pour } i = 0, \ldots, k-1, \tag{I.1}$$

la suite de polynômes $\{P_k\}_{k\geqslant 0}$, appelée famille, est dite orthogonale par rapport à la fonctionnelle linéaire c. Par extension, on dira que le polynôme P_k est orthogonal par rapport à c s'il vérifie les conditions d'orthogonalité (I.1).

Les polynômes orthogonaux étant généralement associés à une matrice de Hankel, il semble nécessaire d'en rappeler ici la définition.

Définition 1.3 – *Matrice de Hankel*
La matrice carrée de dimension k définie par

$$\begin{pmatrix} h_1 & h_2 & \cdots & h_k \\ h_2 & h_3 & \iddots & h_{k+1} \\ \vdots & \iddots & & \vdots \\ h_k & h_{k+1} & \cdots & h_{2k-1} \end{pmatrix}$$

est appelée matrice de Hankel d'ordre k et sera dite générée par le vecteur $(h_1, h_2, \ldots, h_{2k-1})^{\mathrm{T}}$.

Définition 1.4 – *Polynômes orthogonaux réguliers (Draux [32])*
Soit P_k un polynôme de degré k orthogonal par rapport à c. Il est dit régulier si le déterminant de la matrice de Hankel générée par $(c_0, c_1, \ldots, c_{2k-2})^{\mathrm{T}}$ *est non nul. Ce déterminant sera noté* $H_k^{(0)}$.

Nous supposerons dans toute la suite que tous les polynômes orthogonaux considérés sont réguliers (la fonctionnelle c est alors dite définie).

Définition 1.5 – *Matrice de Vandermonde*
La matrice carrée de dimension k définie par

$$\begin{pmatrix} 1 & 1 & \cdots & 1 \\ v_1 & v_2 & \cdots & v_k \\ v_1^2 & v_2^2 & & v_k^2 \\ \vdots & \vdots & & \vdots \\ v_1^{k-1} & v_2^{k-1} & \cdots & v_k^{k-1} \end{pmatrix}$$

est appelée matrice de Vandermonde d'ordre k et sera dite générée par le vecteur $(v_1, v_2, \ldots, v_k)^{\mathrm{T}}$.

Il est à noter que, dans la littérature, on trouve parfois la définition matrice de Vandermonde pour la transposée de la matrice de la Définition 1.5.

Les matrices de Hessenberg jouant un rôle tout particulier dans l'interprétation matricielle des polynômes biorthogonaux, introduisons ici leur définition.

Définition 1.6 – *Matrice de Hessenberg inférieure*
La matrice carrée $H = (h_{i,j})_{\substack{1 \leqslant i \leqslant k \\ 1 \leqslant j \leqslant k}}$ *de dimension k telle que*

$$h_{i,j} = 0 \text{ si } j > i+1$$

est appelée matrice de Hessenberg inférieure.

On peut également définir des matrices de Hessenberg supérieures mais elles n'interviendront pas dans la suite de ce travail.

L'interprétation matricielle des polynômes orthogonaux utilise quant à elle la notion de matrice de Jacobi, que nous définissons.

Définition 1.7 - *Matrice de Jacobi*
Une matrice tridiagonale est appelée matrice de Jacobi.

Tout polynôme unitaire peut être associé à une matrice, appelée matrice compagnon du polynôme en question. Une telle matrice jouera un rôle tout particulier dans la suite. Introduisons dès lors sa définition.

Définition 1.8 - *Matrice compagnon*
Soit P_k un polynôme unitaire de degré k défini par

$$P_k(x) = x^k + p_{k-1}x^{k-1} + \ldots + p_1 x + p_0.$$

La matrice carrée de dimension k définie par

$$\begin{pmatrix} 0 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & 0 & 0 & 1 \\ -p_0 & -p_1 & -p_2 & \cdots & -p_{k-2} & -p_{k-1} \end{pmatrix}$$

est appelée matrice compagnon de P_k.

Elle est souvent notée $\boldsymbol{F}_k$ dans la mesure où ce type de matrice est également rencontré sous le nom de matrice de **F**robenius.

Les polynômes étudiés étant basés sur certaines expressions matricielles, nous parlerons par la suite de matrice principale d'ordre k.

Définition 1.9 - *Matrice principale d'ordre k*
On appelle matrice principale d'ordre k d'une matrice $\boldsymbol{M}$ la sous-matrice, notée $\boldsymbol{M}_k$, composée des k premières lignes et des k premières colonnes de la matrice $\boldsymbol{M}$.

1.2 Relations matricielles et relations de récurrence

Après avoir introduit les définitions de la sous-section 1.1, nous allons maintenant considérer les relations matricielles des polynômes orthogonaux. Même si l'identité de Christoffel-Darboux n'est pas une relation matricielle, nous la classerons tout de même dans cette Section. La justification d'une telle classification sera donnée dans la Partie II.

Les polynômes orthogonaux étant définis par rapport à une fonctionnelle c, il est possible de donner une deuxième définition de ces derniers, sous forme de rapport de déterminants. Plus maniable, elle est souvent utilisée pour des démonstrations théoriques.

Théorème 1.1

Étant donnée une fonctionnelle linéaire c définie par ses moments c_i, $i = 0, 1, \ldots$, alors le polynôme P_k défini par

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

est l'unique polynôme orthogonal unitaire par rapport à c. On rappelle que le dénominateur de cette fraction est noté $H_k^{(0)}$.

Le Théorème suivant est un résultat majeur sur les polynômes orthogonaux unitaires et est parfois considéré comme définition de ces derniers, via le Théorème de Favart-Shohat.

Théorème 1.2 – *(Brezinski [4])*

Les polynômes orthogonaux réguliers unitaires vérifient la relation de récurrence

$$P_{k+1}(x) = (x - \alpha_k)P_k(x) - \beta_k P_{k-1}(x) \text{ pour } k = 0, 1, \ldots \tag{I.2}$$

avec $P_0(x) = 1$ et $P_{-1}(x) = 0$ et

$$\begin{aligned} \beta_0 &= 0 \\ \beta_k &= \frac{c(x^k P_k)}{c(x^{k-1} P_{k-1})} \\ \alpha_k &= \frac{c(x^{k+1} P_k) - \beta_k c(x^k P_{k-1})}{c(x^k P_k)}. \end{aligned}$$

Cette relation est appelée relation de récurrence à trois termes.

On peut montrer (Brezinski [4]) que $c(x^k P_k) = \dfrac{H_{k+1}^{(0)}}{H_k^{(0)}}$. Nous n'avons considéré ici que le cas où tous les polynômes orthogonaux existent. Dans le cas où cette

hypothèse n'est pas satisfaite, on pourra consulter Draux [32] afin d'avoir des relations plus générales.

L'un des résultats les plus importants en ce qui concerne les polynômes orthogonaux est sans doute l'identité de Christoffel-Darboux, que nous rappelons dans le Théorème suivant.

Théorème 1.3 - *(Brezinski [4])*
Pour tout $k \geqslant 0$,

$$\frac{1}{h_k}\left[P_{k+1}(x)P_k(t) - P_{k+1}(t)P_k(x)\right] = (x-t)\sum_{i=0}^{k}\frac{1}{h_i}P_i(x)P_i(t)$$

avec $h_k = c(x^k P_k)$.

Cette relation est connue sous le nom d'identité de Christoffel-Darboux.

Nous allons voir maintenant les relations matricielles concernant les polynômes orthogonaux. Tout d'abord, nous nous intéressons à une autre définition des polynômes orthogonaux, basée sur la relation à trois termes que ceux-ci vérifient.

Théorème 1.4 - *(Brezinski [4])*
Soit $\boldsymbol{J}_k$ *la matrice de Jacobi définie par*

$$\boldsymbol{J}_k = \begin{pmatrix} \alpha_0 & 1 & 0 & \cdots & 0 \\ \beta_1 & \alpha_1 & 1 & \ddots & \vdots \\ 0 & \beta_2 & \alpha_2 & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & 1 \\ 0 & \cdots & 0 & \beta_{k-1} & \alpha_{k-1} \end{pmatrix}$$

où les α_i, $i = 0, 1, \ldots, k-1$ *et les* β_i, $i = 1, 2, \ldots, k-1$ *sont les coefficients qui interviennent dans les relations de récurrence du Théorème 1.2.*
Alors,

$$P_k(x) = |x\boldsymbol{I}_k - \boldsymbol{J}_k|$$

où $\boldsymbol{I}_k$ *désigne naturellement la matrice identité de dimension* k.

Le résultat suivant est alors évident.

Corollaire 1.4.1 - *(Brezinski [4])*
Les zéros de P_k *sont les valeurs propres de* $\boldsymbol{J}_k$.

Notons désormais

$$P_k(x) = x^k + p_{k-1}^{(k)} x^{k-1} + \ldots + p_1^{(k)} x + p_0^{(k)}$$

lorsque le polynôme P_k est unitaire.

Soit alors $\boldsymbol{T}_k$ la matrice triangulaire formée des coefficients des k premiers termes de la suite de polynômes orthogonaux unitaires P_k, c'est-à-dire,

$$\boldsymbol{T}_k = \begin{pmatrix} 1 & 0 & \cdots & 0 & 0 \\ p_0^{(1)} & 1 & \cdots & 0 & 0 \\ p_0^{(2)} & p_1^{(2)} & \cdots & 0 & 0 \\ \vdots & & \ddots & & \vdots \\ p_0^{(k-1)} & p_1^{(k-1)} & \cdots & p_{k-2}^{(k-1)} & 1 \end{pmatrix}.$$

Soit $\boldsymbol{F}_k$ la matrice compagnon de P_k et $\boldsymbol{J}_k$ la matrice définie au Théorème 1.4. Alors, nous pouvons montrer (Gragg [41]) que les matrices $\boldsymbol{J}_k$, $\boldsymbol{T}_k$ et $\boldsymbol{F}_k$ vérifient

$$\boldsymbol{J}_k \boldsymbol{T}_k = \boldsymbol{T}_k \boldsymbol{F}_k.$$

Soit maintenant $\boldsymbol{Z}_k$ la matrice diagonale $\boldsymbol{Z}_k = \mathrm{diag}(x_1^{(k)}, x_2^{(k)}, \ldots, x_k^{(k)})$ où $x_i^{(k)}$ est le i-ème zéro de P_k, les racines du polynôme P_k étant prises avec leur ordre de multiplicité.

Soit de plus $\boldsymbol{V}_k$ la matrice de Vandermonde générée par $\left(x_1^{(k)}, x_2^{(k)}, \ldots, x_k^{(k)}\right)^{\mathrm{T}}$. Alors (Brezinski [4]),

$$\boldsymbol{F}_k \boldsymbol{V}_k = \boldsymbol{V}_k \boldsymbol{Z}_k. \tag{I.3}$$

Donc, les valeurs propres de $\boldsymbol{F}_k$ aussi sont les zéros de P_k.

Ainsi, en posant $\boldsymbol{Q}_k = \boldsymbol{T}_k \boldsymbol{V}_k$, on obtient le

Théorème 1.5 – *(Brezinski [4])*
Les matrices $\boldsymbol{J}_k$ et $\boldsymbol{Z}_k$ sont semblables et

$$\boldsymbol{J}_k \boldsymbol{Q}_k = \boldsymbol{Q}_k \boldsymbol{Z}_k.$$

Les premières relations matricielles étant énoncées, nous devons, pour aller plus en avant, introduire la notion de noyau reproduisant. Certaines relations matricielles en découlent en effet.

Définition 1.10 – *Noyau reproduisant*
La fonction symétrique définie par

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

est appelée noyau reproduisant d'ordre k de la famille des polynômes orthogonaux $\{P_k\}_{k \geqslant 0}$. Les quantités h_i représentent toujours les valeurs $c(x^i P_i)$.

Le résultat suivant est la propriété majeure du noyau reproduisant et justifie son appellation.

Propriété 1.1 – *(Gragg [41])*
Le noyau reproduisant K_k défini dans la Définition 1.10 vérifie

$$c(p(x)K_k(x,t)) = p(t)$$

pour tout polynôme p de degré $\leqslant k$ (c agit sur x et t est un paramètre).

Utilisons alors ce noyau reproduisant afin d'obtenir des relations matricielles supplémentaires concernant les polynômes orthogonaux.

Soit $\boldsymbol{W}_k$ la matrice, définie à partir du noyau reproduisant par les éléments qu'elle contient, par

$$\boldsymbol{W}_k = \left(K_k(x_i^{(k)}, x_j^{(k)})\right)_{\substack{1\leqslant i\leqslant k\\ 1\leqslant j\leqslant k}}$$

Soit de plus $\boldsymbol{H}_k$ la matrice diagonale définie par $\boldsymbol{H}_k = \mathrm{diag}(h_1, h_2, \ldots, h_k)$ où les quantités h_i sont données par $h_i = c(x^i P_i)$.

Alors, on montre aisément la

Proposition 1.1 – *(Brezinski [4])*
Les matrices $\boldsymbol{T}_k$, $\boldsymbol{C}_k$ et $\boldsymbol{H}_k$ vérifient

$$\boldsymbol{T}_k \boldsymbol{C}_k \boldsymbol{T}_k^{\mathrm{T}} = \boldsymbol{H}_k,$$

$\boldsymbol{C}_k$ étant la matrice de Hankel composée des moments de la fonctionnelle c.

Si de plus on pose $\boldsymbol{Q}_k = \boldsymbol{T}_k \boldsymbol{V}_k$, alors on obtient la

Proposition 1.2 – *(Brezinski [4])*
Les matrices $\boldsymbol{H}_k$, $\boldsymbol{Q}_k$ et $\boldsymbol{W}_k$ vérifient

$$\boldsymbol{H}_k = \boldsymbol{Q}_k \boldsymbol{W}_k^{-1} \boldsymbol{Q}_k^{\mathrm{T}}.$$

Des deux dernières Propositions, on trouve un résultat de congruence, que nous donnons dans le

Théorème 1.6 – *(Brezinski [4])*
Les matrices $\boldsymbol{C}_k$, $\boldsymbol{H}_k$ et $\boldsymbol{W}_k^{-1}$ sont congruentes et

$$\boldsymbol{C}_k = \boldsymbol{V}_k \boldsymbol{W}_k^{-1} \boldsymbol{V}_k^{\mathrm{T}}.$$

Enfin, les derniers résultats matriciels sont obtenus si l'on considère la notion de matrice résolvante, dont la définition est la suivante.

Définition 1.11 – *Matrice résolvante*

Soit $\boldsymbol{M}_k$ une matrice de dimension k quelconque. Alors la matrice $\boldsymbol{R}_k$, dépendant de x, définie par

$$\boldsymbol{R}_k(x) = (x\boldsymbol{I}_k - \boldsymbol{M}_k)^{-1}$$

est appelée matrice résolvante de $\boldsymbol{M}_k$.

On rencontre parfois dans certains ouvrages cette même notion pour la matrice définie par $x^{-1}\boldsymbol{R}_k(x^{-1})$.

Ainsi, si $\boldsymbol{H}_k$ est une matrice de Hankel et $\boldsymbol{R}_k(x)$ la matrice résolvante associée, alors on a la décomposition (Gragg [41])

$$\boldsymbol{R}_k(x) = \sum_{i=1}^{k} (x - x_i^{(k)})^{-1} \boldsymbol{R}_i^{(k)}$$

où les matrices $\boldsymbol{R}_i^{(k)}$ vérifient

$$\boldsymbol{I}_k = \sum_{i=1}^{k} \boldsymbol{R}_i^{(k)}$$

et

$$\boldsymbol{J}_k = \sum_{i=1}^{k} \boldsymbol{R}_i^{(k)} x_i^{(k)}.$$

Les coefficients $x_i^{(k)}$ représentent ici les racines du k–ème polynôme orthogonal relatif à la forme linéaire dont les premiers moments génèrent $\boldsymbol{H}_k$.

Si $\boldsymbol{R}_k(x) = \left(r_{i,j}^{(k)}(x)\right)_{\substack{1 \leqslant i \leqslant k \\ 1 \leqslant j \leqslant k}}$, alors on obtient (Brezinski [4]) une expression des coefficients de $\boldsymbol{R}_k$ par

$$r_{i,j}^{(k)}(x) = \frac{1}{h_{j-1}} \sum_{n=1}^{k} \frac{1}{x - x_n^{(k)}} P_{i-1}(x_n^{(k)}) P_{j-1}(x_n^{(k)}) \text{ pour } 1 \leqslant i, j \leqslant k.$$

Tandis que les éléments des matrices $\boldsymbol{R}_n^{(k)}$ sont définis par

$$h_{j-1} P_{i-1}(x_n^{(k)}) P_{j-1}(x_n^{(k)}) \text{ pour } i, j = 1, \ldots, k.$$

On peut ainsi montrer (Gragg [41]) que les matrices $\boldsymbol{R}_n^{(k)}$ sont des projecteurs. À ce titre, elles vérifient

$$\boldsymbol{R}_n^{(k)2} = \boldsymbol{R}_n^{(k)}.$$

Toutes ces relations sont ainsi les principales relations matricielles que l'on peut trouver concernant les polynômes orthogonaux.

2 Polynômes vectoriellement orthogonaux

Après avoir énoncé les principales relations matricielles connues sur les polynômes orthogonaux, intéressons-nous à une première généralisation de ces derniers : les polynômes orthogonaux de dimension $d > 1$. Introduits par Van Iseghem dans [73], ils ont une connexion directe avec les approximants de Padé vectoriels (via leur dénominateur). Nous ne nous intéresserons, dans cette Section, qu'à leurs propriétés intrinsèques, indépendamment de leur implication dans l'algorithme QD vectoriel. Nous verrons notamment les relations matricielles principalement démontrées par Van Iseghem [73, 74, 75].

Dans une première **sous-section**, quelques définitions supplémentaires seront introduites, propres aux polynômes orthogonaux de dimension $d > 1$.

Dans la **seconde sous-section**, certaines relations concernant les polynômes vectoriellement orthogonaux seront rappelées.

2.1 Définitions

L'orthogonalité vectorielle fait appel à des notions plus complexes que celles rencontrées dans le cadre des polynômes orthogonaux. Il est utile d'en définir quelques unes ici.

Définition 2.1 – *Fonctionnelle linéaire vectorielle*
La fonction C définie de $\mathbb{C}[X]$ dans $\mathbb{C}^d$, où $d > 1$, par

$$C(x^i) = C_i \text{ pour } i \geqslant 0$$

est appelée fonctionnelle linéaire vectorielle (de dimension d). Les quantités C_i sont bien sûr des vecteurs de $\mathbb{C}^d$.

La fonctionnelle C sera alors définie par ses composantes

$$C = \left(C^{(1)}, \cdots, C^{(d)}\right)^{\mathrm{T}}$$

où $C^{(j)}$ associe à x^i la j–ème composante du vecteur C_i.

On définit, à partir de ces fonctionnelles vectorielles les polynômes orthogonaux de dimension $d > 1$.

Définition 2.2 – *Polynômes vectoriellement orthogonaux (Van Iseghem [73])*
On appelle polynômes orthogonaux de dimension $d > 1$ par rapport à la fonctionnelle linéaire vectorielle C ou polynômes vectoriellement orthogonaux par rapport à C toute suite de polynômes $\{P_r\}_{r \geqslant 0}$ de $\mathbb{C}[X]$ vérifiant les relations d'orthogonalité

$$r = nd + k, \quad \begin{cases} C(x^i P_r(x)) = 0 \text{ pour } 0 \leqslant i \leqslant n-1 \\ C^{(\alpha)}(x^n P_r(x)) = 0 \text{ pour } 1 \leqslant \alpha \leqslant k \end{cases} \tag{I.4}$$

où l'on a effectué la division euclidienne de r par d.

Lorsque l'on cherche à trouver le polynôme P_r satisfaisant les conditions d'orthogonalité (I.4), on considère une matrice de Hankel généralisée et le déterminant qui lui est associé.

Définition 2.3 - *Matrice de Hankel généralisée*
On appelle matrice de Hankel généralisée la matrice issue du système linéaire (I.4), c'est à dire celle qui s'écrit

$$\begin{pmatrix} C_0 & C_1 & \cdots & C_{r-1} \\ C_1 & C_2 & \cdots & C_r \\ \vdots & \vdots & & \vdots \\ C_{n-1} & C_n & \cdots & C_{n+r-2} \\ C_n^{(k)} & C_{n+1}^{(k)} & \cdots & C_{n+r-1}^{(k)} \end{pmatrix}$$

où $r = nd+k$, les quantités C_i sont des vecteurs de $\mathbb{C}^d$, et les $C_i^{(k)}$ sont les vecteurs de $\mathbb{C}^k$ formés par les k premières composantes des vecteurs C_i. La matrice ainsi considérée est carrée.

Ces notions sont les principales relations spécifiques à la définition des polynômes vectoriellement orthogonaux et peuvent être mises en parallèle avec celles concernant les polynômes orthogonaux de la Section 1.

2.2 Relations de récurrence

Voyons dans cette sous-section les relations matricielles et de récurrence qui existent pour les polynômes vectoriellement orthogonaux. Elles sont, pour la plupart, une généralisation de celles existant pour les polynômes orthogonaux.

Ainsi, on trouve une expression sous forme de déterminant pour les polynômes vectoriellement orthogonaux. Lorsque les déterminants des matrices de Hankel généralisées sont tous non nuls, alors la famille des polynômes orthogonaux de dimension $d > 1$ existe et l'on a le

Théorème 2.1 - *(Van Iseghem [73])*
Le polynôme P_r, où $r = nd + k$ défini par

$$P_r(x) = \frac{\begin{vmatrix} C_0 & \cdots & C_r \\ \vdots & & \vdots \\ C_{n-1} & \cdots & C_{n+r-1} \\ C_n^{(k)} & \cdots & C_{n+r}^{(k)} \\ 1 & \cdots & x^r \end{vmatrix}}{\begin{vmatrix} C_0 & \cdots & C_{r-1} \\ \vdots & & \vdots \\ C_{n-1} & \cdots & C_{n+r-2} \\ C_n^{(k)} & \cdots & C_{n+r-1}^{(k)} \end{vmatrix}}$$

est vectoriellement orthogonal par rapport à la fonctionnelle C. Ainsi défini, ce polynôme est unitaire et unique.

Ainsi, de façon analogue aux polynômes orthogonaux, l'existence de la suite des polynômes vectoriellement orthogonaux est subordonnée à la non nullité des déterminants de Hankel (généralisés cette fois) introduits dans la Définition 2.3, ce que nous supposerons désormais.

Le Théorème suivant est, comme pour les polynômes orthogonaux classiques, un résultat très important pour les polynômes orthogonaux de dimension $d > 1$. Il est également parfois utilisé pour définir ces derniers via une extension du Théorème de Favart-Shohat démontré par Van Iseghem [73].

Théorème 2.2 - *(Van Iseghem [73])*
Les polynômes vectoriellement orthogonaux unitaires vérifient une relation de récurrence à $d+2$ termes du type

$$P_{r+1}(x) = (x - \beta_r)P_r(x) + \sum_{i=1}^{d} \alpha_i^{(r)} P_{r-i}(x) \text{ pour } r = 0, 1, \cdots \tag{I.5}$$

avec $P_0(x) = 1$ et $P_i(x) = 0$ pour $i < 0$.

Les coefficients $\alpha_i^{(r)}$ pour $i = d-k, \ldots, d$ sont obtenus en multipliant (I.5) par x^{n-1} puis en appliquant $C^{(k+1)}, \ldots, C^{(d)}$ successivement et en utilisant les conditions d'orthogonalité (I.4). On obtient ainsi l'expression générale des coefficients $\alpha_i^{(r)}$ pour $i = d-k, \ldots, d$ après résolution d'un système triangulaire.

D'autre part, on multiplie (I.5) par x^n et l'on applique $C^{(1)}, \ldots, C^{(k)}$. Les conditions d'orthogonalité sont à nouveau utilisées pour l'obtention des coefficients $\alpha_i^{(r)}$ pour $i = 1, \ldots, d-k-1$. Ces derniers dépendent des $\alpha_i^{(r)}$ pour $i = d-k, \ldots, d$.

Le coefficient β_r est quant à lui obtenu en multipliant l'égalité (I.5) par x^n et en applicant $C^{(k+1)}$. Les conditions d'orthogonalité nous permettent à nouveau d'obtenir l'expression de ce coefficient, qui dépend des $\alpha_i^{(r)}$.

Remarque 2.1

On ne donnera pas ici une expression explicite de tous ces coefficients dans la mesure où celle-ci est assez lourde et n'est, de plus, pas utilisée dans la suite. Pour plus de détails, on pourra se référer à [73].

Remarque 2.2

Une identité de type Christoffel-Darboux peut être trouvée dans [77] mais elle utilise une forme bilinéaire différente de celle que nous allons utiliser dans la Deuxième Partie. Nous ne la rappellerons donc pas ici.

Conclusion

Des deux Sections précédentes, on remarque que certaines des notions connues pour les polynômes orthogonaux ont leur prolongement pour les polynômes orthogonaux de dimension $d > 1$ (relations de récurrence, expression sous forme de déterminants, conditions d'orthogonalité ...).

De plus, l'interprétation matricielle des polynômes vectoriellement orthogonaux n'a pas encore été traitée dans le détail. À travers la biorthogonalité, certains résultats matriciels sur les polynômes vectoriellement orthogonaux seront ainsi démontrés ou retrouvés.

Deuxième partie
Interprétation matricielle des polynômes biorthogonaux

Introduction

Le but de cette partie est d'étudier dans le détail les polynômes biorthogonaux et, d'une certaine manière, d'étendre certains des résultats généraux déjà connus sur les polynômes orthogonaux (voir [4, 8, 32]) ainsi que certains des résultats sur les polynômes orthogonaux de dimension $d > 1$ (voir [76]), notamment en terme matriciel. Nous nous intéresserons donc tout naturellement aux relations de récurrence que l'on peut trouver ainsi qu'aux éventuelles propriétés quant aux zéros de tels polynômes. Les notions telles que noyau reproduisant et identité de type Christoffel-Darboux auront tout naturellement leur prolongement ici.

C'est ainsi que dans la **première section** nous allons tout d'abord définir la biorthogonalité ainsi que les conditions d'existence et d'unicité des polynômes biorthogonaux. Des relations de récurrence seront également données.

Dans la **deuxième section**, des relations matricielles seront mises en évidence, notamment des relations utilisant les zéros des polynômes biorthogonaux. Les valeurs propres de certaines matrices seront également considérées. Dans ce cadre, les matrices de Hessenberg et de Vandermonde joueront un rôle essentiel. D'autre part, l'introduction de polynômes biorthogonaux à droite et polynômes biorthogonaux à gauche permettra quant à elle diverses relations matricielles.

La **troisième section** sera consacrée à des notions telles que noyau reproduisant et relations de type Christoffel-Darboux. Ainsi, en étudiant les zéros des polynômes biorthogonaux, de nouvelles relations matricielles vont apparaître. Ces relations permettront l'écriture d'identités de type Christoffel-Darboux.

La **quatrième section** considérera une classe de projecteurs orthogonaux issus des relations matricielles des Sections précédentes. Ces projecteurs seront tout naturellement une extension des projecteurs connus sur les polynômes orthogonaux.

Enfin, dans la **cinquième section**, on montrera la relation qui existe entre le calcul des polynômes biorthogonaux et la méthode de bordage. Les cas particuliers des polynômes orthogonaux et des polynômes orthogonaux de dimension -1 seront étudiés.

1 Biorthogonalité

Les polynômes vectoriellement orthogonaux sont une extension des polynômes orthogonaux (si l'on considère par exemple les relations de récurrence qu'ils vérifient). Une notion plus large encore est celle des polynômes biorthogonaux, à laquelle nous consacrons cette Section.

Dans la **première sous-section**, quelques définitions concernant la biorthogonalité seront introduites. Elles seront nécessaires à la compréhension de cette dernière. La notion de polynômes biorthogonaux sera bien entendu abordée.

Les polynômes biorthogonaux définis, il sera alors possible d'étudier leur existence ainsi que leur éventuelle unicité dans la **deuxième sous-section.**

Enfin, quelques relations de récurrence concernant ces polynômes biorthogonaux seront considérées dans la **troisième sous-section.** Ces relations de récurrence seront utiles pour d'éventuelles relations matricielles démontrées dans la Section 2.

La notion de biorthogonalité étudiée dans cette partie est celle qui a été définie par Brezinski dans [8].

1.1 Définitions

La biorthogonalité est une notion qui concerne d'une part des formes linéaires et d'autre part certains polynômes. Introduisons alors ici les définitions nécessaires à sa compréhension.

Soient $\{\mathcal{L}_i\}_{i\geqslant 0}$ des fonctionnelles linéaires définies sur $\mathcal{P}$ par

$$\mathcal{L}_i(x^j) = c_{i,j} \text{ pour } i,j = 0,1,\ldots$$

Évidemment, compte tenu de la définition de ces fonctionnelles, il est possible de regrouper les différents coefficients $c_{i,j}$ sous forme matricielle en posant

$$\boldsymbol{L} = \begin{pmatrix} c_{0,0} & c_{0,1} & \cdots \\ c_{1,0} & \ddots & \\ \vdots & & \ddots \end{pmatrix}.$$

Dans le cas où l'on considère la matrice principale d'ordre k de la matrice précédente, celle-ci peut être rencontrée sous le nom de matrice de Gram et est alors notée $[\mathcal{L} : f]$ où f désigne le vecteur $(1, x, x^2, \ldots, x^k)^{\mathrm{T}}$ [78].

De façon analogue, la matrice $\boldsymbol{L}^{\mathrm{T}}$ représentera les formes linéaires $\mathcal{L}_i^{\mathrm{T}}$ définies par

$$\mathcal{L}_i^{\mathrm{T}}(x^j) = c_{j,i} \text{ pour } i,j = 0,1,\ldots$$

Ainsi, pour une matrice $\boldsymbol{L}$ donnée, deux familles de fonctionnelles peuvent être facilement associées (ou tout au moins les moments que de telles formes linéaires peuvent prendre sur les premiers éléments de la base canonique).

On définit la suite de polynômes biorthogonaux $\{P_k\}_{k\geqslant 0}$.

Définition 1.1 - *Polynômes biorthogonaux (Brezinski [8])*
Les polynômes $P_k \in \mathcal{P}_k$ vérifiant

$$\mathcal{L}_i(P_k) = 0 \text{ pour } i = 0, \ldots, k-1 \tag{II.1}$$

sont appelés polynômes biorthogonaux.

On appellera également ces derniers polynômes biorthogonaux à gauche de $\boldsymbol{L}$. De même, on définira les polynômes biorthogonaux à droite de $\boldsymbol{L}$, notés $\widetilde{P}_k$, par

$$\mathcal{L}_i^{\mathrm{T}}(\widetilde{P}_k) = 0 \text{ pour } i = 0, \ldots, k-1. \tag{II.2}$$

Le polynôme P_k (respectivement $\widetilde{P}_k$) sera donc biorthogonal par rapport aux $k-1$ premières formes linéaires $\mathcal{L}_i$ (respectivement $\mathcal{L}_i^{\mathrm{T}}$).

Convention

> *On pourra parler par abus de langage dans la suite, lorsqu'il n'y a pas d'ambiguïté, de polynômes biorthogonaux sans préciser les fonctionnelles utilisées pour cette biorthogonalité.*

À partir de la Définition 1.9 de la Première Partie, on introduit la notion de matrice fortement régulière que l'on définit ci-après.

Définition 1.2 - *Matrice fortement régulière*
La matrice carrée $\boldsymbol{M}_k$ de dimension k sera dite fortement régulière si ses k premières matrices principales d'ordre k sont inversibles (ont un déterminant non nul).

1.2 Existence, unicité des polynômes biorthogonaux

Il faut désormais considérer l'existence et l'unicité des suites de polynômes P_k et $\widetilde{P}_k$ définies plus haut.

Nous allons donc tout d'abord considérer l'existence de telles suites avant de s'interroger sur leur unicité.

1.2.1 Existence

Pour étudier l'existence des polynômes P_k et $\widetilde{P}_k$, il faut s'intéresser aux diverses manières de les déterminer, d'après les conditions qu'ils doivent remplir selon (II.1) et (II.2).

Pour cela, notons désormais, de façon canonique,

$$P_k(x) = \sum_{j=0}^{k} p_k^{(j)} x^j \text{ et } \widetilde{P}_k(x) = \sum_{j=0}^{k} \widetilde{p}_k^{(j)} x^j. \tag{II.3}$$

On vérifie aisément que le polynôme défini par

$$\begin{vmatrix} c_{0,0} & \cdots & \cdots & c_{0,k} \\ \vdots & & & \vdots \\ c_{k-1,0} & \cdots & \cdots & c_{k-1,k} \\ 1 & x & \cdots & x^k \end{vmatrix} \tag{II.4}$$

satisfait les conditions d'orthogonalité (II.1). Il suffit, pour cela, d'appliquer $\mathcal{L}_i$ pour $i = 0, \ldots, k-1$ et de constater que deux lignes sont identiques. Le déterminant est alors nul.

Un tel polynôme existera donc toujours. Un problème peut se poser si l'on veut qu'en plus ce polynôme soit de degré k exactement. Alors, il faudra de plus que

$$|\boldsymbol{L}_k| = \begin{vmatrix} c_{0,0} & \cdots & c_{0,k-1} \\ \vdots & & \vdots \\ c_{k-1,0} & \cdots & c_{k-1,k-1} \end{vmatrix} \neq 0. \tag{II.5}$$

En effet, $|\boldsymbol{L}_k|$ correspond au coefficient de plus haut degré du polynôme défini en (II.4). Nous supposerons cette condition désormais satisfaite pour tout $k \geqslant 0$.

1.2.2 Unicité

Supposons alors l'existence d'un polynôme de degré k exactement vérifiant les égalités (II.1). Interrogeons-nous alors sur son unicité.

En divisant P_k par $p_k^{(k)} = (-1)^k |\boldsymbol{L}_k|$, les équations (II.1) deviennent

$$\sum_{j=0}^{k-1} \frac{p_k^{(j)}}{p_k^{(k)}} c_{i,j} = -c_{i,k} \text{ pour } i = 0,1,\ldots,k-1,$$

où les inconnues sont bien entendu $p_k^{(0)}, p_k^{(1)}, \ldots, p_k^{(k-1)}$.

Ce système étant un système de k équations à k inconnues, il admet une solution unique si et seulement si il est de Cramer, c'est-à-dire s'il admet un déterminant non nul. Or, le déterminant de ce système est

$$\frac{1}{p_k^{(k)}} \begin{vmatrix} c_{0,0} & c_{0,1} & \cdots & c_{0,k-1} \\ c_{1,0} & c_{1,1} & & c_{1,k-1} \\ \vdots & \vdots & & \vdots \\ c_{k-1,0} & c_{k-1,1} & \cdots & c_{k-1,k-1} \end{vmatrix} = \frac{|\boldsymbol{L}_k|}{p_k^{(k)}}.$$

De plus, pour pouvoir diviser par $p_k^{(k)}$, il faut que celui-ci soit non nul, c'est-à-dire

$$|\boldsymbol{L}_k| \neq 0.$$

Ainsi, le polynôme P_k unitaire de degré k exactement existe et est unique si la condition (II.5) est remplie. Et il s'écrit

$$P_k(x) = \frac{\begin{vmatrix} c_{0,0} & \cdots & \cdots & c_{0,k} \\ \vdots & & & \vdots \\ c_{k-1,0} & \cdots & \cdots & c_{k-1,k} \\ 1 & x & \cdots & x^k \end{vmatrix}}{|\boldsymbol{L}_k|}. \tag{II.6}$$

On retrouve tout naturellement une extension des définitions des polynômes orthogonaux et vectoriellement orthogonaux de la Première Partie.

De même manière, on montre que le polynôme unitaire $\widetilde{P}_k$ de degré k exactement existe et est unique si et seulement si $|\boldsymbol{L}_k^{\mathrm{T}}| \neq 0$. Comme $|\boldsymbol{L}_k^{\mathrm{T}}| = |\boldsymbol{L}_k|$, la condition d'existence et d'unicité est identique pour les deux polynômes. Nous supposerons dès lors que $|\boldsymbol{L}_k| \neq 0$ pour tout $k \geqslant 0$.

1.3 Relations de récurrence

Nous allons étudier les diverses relations de récurrence que l'on peut exprimer pour les polynômes P_k et $\widetilde{P}_k$.

Deux types de relations seront considérées ici. Tout d'abord, nous aborderons les relations où une seule famille de polynômes est utilisée. Ces relations seront appelées relations fermées. Ensuite, des relations seront utilisées où les deux familles sont considérées simultanément. Ces relations seront, quant à elles, nommées relations mixtes.

1.3.1 Relations fermées

On appellera relations fermées des relations qui ne prennent en compte que les polynômes d'une seule famille.

On rappelle que $|\boldsymbol{L}_k| \neq 0$ pour tout k (c'est-à-dire que tous les polynômes biorthogonaux existent et sont de degré k exactement). Alors, deux principales façons d'écrire P_k et $\widetilde{P}_k$ peuvent être considérées. Elles figurent dans les Propositions (1.1) et (1.3).

Proposition 1.1

Les polynômes P_k s'écrivent

$$P_{k+1}(x) = xP_k(x) - \sum_{j=0}^{k} a_k^{(j)} P_j(x) \text{ pour } k = 0, 1, \ldots \tag{II.7}$$

avec $P_0(x) = 1$ et où les coefficients $a_k^{(j)}$ sont donnés par

$$\begin{aligned} a_k^{(0)} &= \frac{\mathcal{L}_0(xP_k)}{\mathcal{L}_0(P_0)} \\ a_k^{(j)} &= \frac{\mathcal{L}_j(xP_k)}{\mathcal{L}_j(P_j)} - \sum_{l=0}^{j-1} a_k^{(l)} \frac{\mathcal{L}_j(P_l)}{\mathcal{L}_j(P_j)}. \end{aligned}$$

Preuve :

Les polynômes P_j étant de degré j exactement, alors les polynômes xP_j, P_j, $\ldots$, P_0 forment une base de $\mathcal{P}_{j+1}$. L'écriture (II.7) est alors possible et unique.

Pour l'expression des coefficients, il suffit d'appliquer $\mathcal{L}_j$ à P_{k+1} successivement pour $\mathcal{L}_0, \mathcal{L}_1, \ldots, \mathcal{L}_k$. Le résultat devant être nul d'après les conditions d'orthogonalité (II.1), on trouve le résultat.

■

Comme dans le cas des polynômes orthogonaux ou des polynômes orthogonaux de dimension $d > 1$, une telle expression pose un problème si la valeur de $\mathcal{L}_j(P_j)$ est nulle.

Or, on démontre la

Proposition 1.2

Soit P_k le polynôme biorthogonal unitaire par rapport aux formes linéaires $\mathcal{L}_i$. Supposons la matrice $\boldsymbol{L}_{k+1}$ fortement régulière.

Alors les polynômes P_i pour $0 \leqslant i \leqslant k$ existent et

$$\mathcal{L}_k(P_k) = \frac{|\boldsymbol{L}_{k+1}|}{|\boldsymbol{L}_k|}.$$

Preuve :

Il suffit de considérer P_k sous la forme de rapport de déterminants exprimé en (II.6), ce qui est possible puisque $|\boldsymbol{L}_k| \neq 0$. On applique alors $\mathcal{L}_k$ et le résultat est immédiat en revenant à la définition de cette forme linéaire.

■

Ainsi, dans l'hypothèse où les déterminants $|\boldsymbol{L}_k|$ sont tous non nuls, l'expression des coefficients de la Proposition 1.1 a bien un sens. Il s'agit à nouveau d'une généralisation de l'écriture des polynômes orthogonaux et orthogonaux de dimension $d > 1$. La différence majeure est que la relation de récurrence n'a pas un nombre de termes fixé puisque ce dernier dépend du degré du polynôme.

Bien sûr, une expression strictement identique peut être obtenue pour les polynômes biorthogonaux $\widetilde{P}_k$.

Proposition 1.3
Les polynômes $\widetilde{P}_k$ sont exprimés par

$$\widetilde{P}_{k+1}(x) = x\widetilde{P}_k(x) - \sum_{j=0}^{k} \widetilde{a}_k^{(j)} \widetilde{P}_j(x) \tag{II.8}$$

où $\widetilde{P}_0(x) = 1$ et où les coefficients $\widetilde{a}_k^{(j)}$ ont une expression similaire à celle de la Proposition 1.1.

Preuve :

Il suffit simplement d'appliquer $\mathcal{L}_i^{\mathrm{T}}$ et d'utiliser les conditions d'orthogonalité (II.2).

■

Bien qu'évidentes, ces deux expressions joueront un rôle pour l'interprétation matricielle proprement dite dans la Section 2.

1.3.2 Relations mixtes

Après avoir mis en évidence les relations que vérifient les polynômes P_k et $\widetilde{P}_k$ par rapport aux polynômes de la même famille, cherchons maintenant une expression des polynômes P_k et $\widetilde{P}_k$ qui utilise les deux familles de polynômes simultanément.

Ceci nous permettra d'étudier les liens matriciels qui unissent les deux familles de polynômes biorthogonaux.

Proposition 1.4
Les polynômes P_k vérifient la relation

$$P_{k+1}(x) = x\widetilde{P}_k(x) - \sum_{j=0}^{k} b_k^{(j)} P_j(x) \text{ pour } k = 0, 1, \ldots \tag{II.9}$$

avec $P_0(x) = \widetilde{P}_0(x) = 1$

$$\begin{aligned} b_k^{(0)} &= \frac{\mathcal{L}_0(x\widetilde{P}_k)}{\mathcal{L}_0(P_0)} \\ b_k^{(j)} &= \frac{\mathcal{L}_j(x\widetilde{P}_k)}{\mathcal{L}_j(P_j)} - \sum_{l=0}^{j-1} b_k^{(l)} \frac{\mathcal{L}_j(P_l)}{\mathcal{L}_j(P_j)} \text{ pour } j = 1, \ldots, k. \end{aligned}$$

Preuve :

La démonstration de cette Proposition, bien que similaire à celle de la Proposition 1.1, en diffère quelque peu.

L'écriture (II.9) est toujours possible puisque les polynômes $x\widetilde{P}_k(x)$, $P_k(x)$, $P_{k-1}(x), \ldots, P_0(x)$ sont de degré respectif $k+1$, k, ..., 1, 0. Ainsi ils forment une base de $\mathcal{P}_{k+1}$.

Pour trouver $b_k^{(0)}$, on applique $\mathcal{L}_0$ à (II.9). Par orthogonalité (II.1), on obtient $\mathcal{L}_0(P_j) = 0$ si $j \neq 0$ ce qui nous donne l'expression du coefficient.

Pour les autres coefficients, on applique $\mathcal{L}_i$ à (II.9) et on utilise à nouveau les conditions d'orthogonalité (II.1). Il est à noter que $\mathcal{L}_i(x\widetilde{P}_k)$ est présent dans tous les coefficients dans la mesure où les fonctionnelles $\mathcal{L}_i$ et les polynômes $\widetilde{P}_k$ ne sont, en général, pas liés.

■

Cette expression est également toujours définie dans la mesure où $\mathcal{L}_j(P_j) \neq 0$ pour tout j (puisque $|\boldsymbol{L}_{j+1}| \neq 0$).

Une formulation analogue peut ainsi être énoncée pour les polynômes $\widetilde{P}_k$.

Proposition 1.5
Les polynômes $\widetilde{P}_k$ s'expriment par

$$\widetilde{P}_{k+1}(x) = xP_k(x) - \sum_{j=0}^{k} \widetilde{b}_k^{(j)} \widetilde{P}_j(x) \tag{II.10}$$

où $P_0(x) = \widetilde{P}_0(x) = 1$ et où les coefficients $\widetilde{b}_k^{(j)}$ ont une expression similaire à celle introduite à la Proposition 1.4.

La relation (II.7) généralise donc naturellement la relation à trois termes des polynômes orthogonaux et celle à $d+2$ termes des polynômes orthogonaux de dimension $d > 1$ étudiés par Van Iseghem.

La relation (II.9), quant à elle, est nouvelle, dans la mesure où ici les deux familles de polynômes sont considérées simultanément, ce qui n'était pas le cas précédemment.

2 Relations matricielles

Après avoir exprimé les différentes façons d'écrire les polynômes P_k et $\widetilde{P}_k$, intéressons-nous désormais aux différentes relations matricielles qui émanent directement des polynômes biorthogonaux et des relations polynomiales définies dans la Section 1.

Pour cela, nous considérerons dans la **première sous-section** des relations générales.

Dans la **deuxième sous-section**, les zéros des polynômes biorthogonaux seront considérés et de nouvelles relations matricielles seront obtenues et démontrées.

Enfin, des relations matricielles seront mises en évidence dans la **troisième sous-section** entre les polynômes biorthogonaux à gauche et les polynômes biorthogonaux à droite.

2.1 Relations générales

Nous allons tout d'abord introduire deux premières matrices issues des définitions même des polynômes biorthogonaux introduites en (II.3).

Posons

$$\boldsymbol{T}_k = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ p_1^{(0)} & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ p_{k-1}^{(0)} & \cdots & p_{k-1}^{(k-2)} & 1 \end{pmatrix}$$

et

$$\widetilde{\boldsymbol{T}}_k = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ \widetilde{p}_1^{(0)} & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ \widetilde{p}_{k-1}^{(0)} & \cdots & \widetilde{p}_{k-1}^{(k-2)} & 1 \end{pmatrix}$$

qui sont donc les matrices des coefficients des k premiers polynômes biorthogonaux respectivement à gauche et à droite.

Énonçons tout d'abord quelques relations ne dépendant uniquement que de la matrice $\boldsymbol{L}_k$ et des matrices $\boldsymbol{T}_k$ et $\widetilde{\boldsymbol{T}}_k$.

Proposition 2.1

Si $\boldsymbol{T}_k$ est la matrice des coefficients des k premiers polynômes biorthogonaux à gauche et $\boldsymbol{L}_k$ la matrice des formes linéaires associées, alors la matrice $\boldsymbol{L}_k\boldsymbol{T}_k^{\mathrm{T}}$ est une matrice triangulaire inférieure.

Preuve :

L'élément de la ligne i, colonne j de ce produit est, par définition, le produit scalaire de la ligne i de $\boldsymbol{L}_k$ par la colonne j de $\boldsymbol{T}_k^{\mathrm{T}}$ (donc la ligne j de $\boldsymbol{T}_k$).

Il s'agit donc de $\mathcal{L}_{i-1}(P_{j-1})$, qui est nul pour $i = 1, 2, \ldots, j-1$ d'après les conditions d'orthogonalité énoncées en (II.1), ce qui montre la forme triangulaire inférieure du produit matriciel considéré.

■

Un raisonnement analogue sur les polynômes biorthogonaux à droite, c'est-à-dire sur les matrices $\widetilde{\boldsymbol{T}}_k$ et $\boldsymbol{L}_k^{\mathrm{T}}$ nous donne la

Proposition 2.2

Si $\widetilde{\boldsymbol{T}}_k$ est la matrice des coefficients des k premiers polynômes biorthogonaux à droite et $\boldsymbol{L}_k$ la matrice des formes linéaires associées, alors la matrice $\widetilde{\boldsymbol{T}}_k\boldsymbol{L}_k$ est une matrice triangulaire supérieure.

Preuve :

Il s'agit, en effet ici, de considérer le produit scalaire de la ligne i de $\widetilde{\boldsymbol{T}}_k$ par la colonne j de $\boldsymbol{L}_k$, c'est-à-dire $\mathcal{L}_{j-1}^{\mathrm{T}}(\widetilde{P}_{i-1})$, qui est nul pour $j = 1, 2, \ldots, i-1$ d'après, cette fois-ci, les conditions d'orthogonalité exprimées en (II.2), ce qui nous donne la forme triangulaire supérieure en question.

■

En combinant les deux Propositions précédentes, on obtient de façon triviale la

Proposition 2.3

Si $\boldsymbol{T}_k$ est la matrice des k premiers polynômes biorthogonaux à gauche, $\widetilde{\boldsymbol{T}}_k$ la matrice des k premiers polynômes biorthogonaux à droite et $\boldsymbol{L}_k$ la matrice des fonctionnelles linéaires associées, alors la matrice $\widetilde{\boldsymbol{T}}_k\boldsymbol{L}_k\boldsymbol{T}_k^{\mathrm{T}}$ est une matrice diagonale.

Preuve :

D'après la Proposition 2.1, et par définition de $\widetilde{\boldsymbol{T}}_k$, $\widetilde{\boldsymbol{T}}_k \boldsymbol{L}_k \boldsymbol{T}_k^{\mathrm{T}}$ est le produit de deux matrices triangulaires inférieures. Elle est donc triangulaire inférieure.

De plus, d'après la Proposition 2.2 et par définition de $\boldsymbol{T}_k^{\mathrm{T}}$, c'est aussi le produit de deux matrices triangulaires supérieures. Elle est triangulaire supérieure.

Elle est donc diagonale.

■

Cette matrice diagonale sera notée $\boldsymbol{D}_k$ et ses éléments diagonaux seront tout naturellement notés $(d_0, \ldots, d_{k-1})$.

On remarque ici que si les matrices $\boldsymbol{T}_k$ et $\widetilde{\boldsymbol{T}}_k$ sont inversibles (ce qui est le cas puisqu'elles sont à diagonale unité), alors

$$\boldsymbol{L}_k = \widetilde{\boldsymbol{T}}_k^{-1} \boldsymbol{D}_k \boldsymbol{T}_k^{-\mathrm{T}}.$$

Or, la matrice $\widetilde{\boldsymbol{T}}_k$ étant triangulaire inférieure et la matrice $\boldsymbol{T}_k$ également, il s'agit en fait de la décomposition LU de la matrice $\boldsymbol{L}_k$. La décomposition LU est donc directement liée aux polynômes biorthogonaux. La matrice diagonale $\boldsymbol{D}_k$ ne servant, en fait, qu'à la normalisation des deux suites de polynômes.

Remarque 2.1

On retrouve ici une généralisation de la décomposition de Choleski d'une matrice symétrique. En effet, pour une matrice symétrique, les polynômes biorthogonaux à gauche et à droite sont identiques (puisque les deux matrices $\boldsymbol{L}_k$ et $\boldsymbol{L}_k^{\mathrm{T}}$ sont alors identiques).

Suite aux écritures des polynômes P_k et $\widetilde{P}_k$ en (II.7) et en (II.8), nous allons définir deux nouvelles matrices de coefficients.

Posons

$$\boldsymbol{J}_k = \begin{pmatrix} a_0^{(0)} & 1 & 0 & \ldots & 0 \\ a_1^{(0)} & a_1^{(1)} & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ \vdots & \vdots & & \ddots & 1 \\ a_{k-1}^{(0)} & a_{k-1}^{(1)} & \ldots & \ldots & a_{k-1}^{(k-1)} \end{pmatrix} \tag{II.11}$$

puis

$$\widetilde{\boldsymbol{J}}_k = \begin{pmatrix} \widetilde{a}_0^{(0)} & 1 & 0 & \dots & 0 \\ \widetilde{a}_1^{(0)} & \widetilde{a}_1^{(1)} & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ \vdots & \vdots & & \ddots & 1 \\ \widetilde{a}_{k-1}^{(0)} & \widetilde{a}_{k-1}^{(1)} & \dots & \dots & \widetilde{a}_{k-1}^{(k-1)} \end{pmatrix} \tag{II.12}$$

qui sont deux matrices de Hessenberg inférieures.

Introduisons de plus la matrice compagnon de P_k.

$$\boldsymbol{F}_k = \begin{pmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & & & \vdots \\ -p_k^{(0)} & -p_k^{(1)} & \dots & -p_k^{(k-2)} & -p_k^{(k-1)} \end{pmatrix}.$$

Alors nous avons la

Proposition 2.4

Si $\boldsymbol{F}_k$ est la matrice compagnon de P_k et $\boldsymbol{J}_k$ la matrice de Hessenberg inférieure introduite en (II.11), alors les matrices $\boldsymbol{F}_k$ et $\boldsymbol{J}_k$ sont semblables et l'on a

$$\boldsymbol{J}_k \boldsymbol{T}_k = \boldsymbol{T}_k \boldsymbol{F}_k. \tag{II.13}$$

Preuve :

Adoptons avant tout quelques conventions.

Désormais,

$$\begin{aligned} p_k^{(k)} &= 1 \\ p_k^{(i)} &= 0 \text{ si } i > k \\ a_i^{(i+1)} &= 1 \\ a_i^{(i+j)} &= 0 \text{ si } j > 1. \end{aligned}$$

Ainsi, $\boldsymbol{T}_k = \left(p_{i-1}^{(j-1)}\right)_{1 \leqslant i,j \leqslant k}$ et $\boldsymbol{J}_k = \left(a_{i-1}^{(j-1)}\right)_{1 \leqslant i,j \leqslant k}$. Alors, l'élément de la ligne i et colonne j du produit $\boldsymbol{J}_k \boldsymbol{T}_k$ est

$$\sum_{l=1}^{k} a_{i-1}^{(l-1)} p_{l-1}^{(j-1)}. \tag{II.14}$$

D'après (II.7), on a

$$xP_k(x) - P_{k+1}(x) = \sum_{j=0}^{k} a_k^{(j)} \sum_{l=0}^{j} p_j^{(l)} x^l = \sum_{j=0}^{k} a_k^{(j)} \sum_{l=0}^{k} p_j^{(l)} x^l$$

en vertu des conventions adoptées plus haut.

Ainsi,

$$\sum_{l=1}^{k+1} (p_k^{(l-1)} - p_{k+1}^{(l)}) x^l - p_{k+1}^{(0)} = \sum_{l=0}^{k} \left(\sum_{j=0}^{k} a_k^{(j)} \right) p_j^{(l)} x^l.$$

Et, par identification des coefficients, on trouve

$$p_{k+1}^{(0)} = -\sum_{j=0}^{k} a_k^{(j)} p_j^{(0)} \tag{II.15}$$

$$p_k^{(l-1)} - p_{k+1}^{(l)} = \sum_{j=0}^{k} a_k^{(j)} p_j^{(l)} \text{ si } l > 1. \tag{II.16}$$

Ainsi, d'après (II.14), la colonne d'indice 1 du produit matriciel $\boldsymbol{J}_k \boldsymbol{T}_k$ pour la ligne i devient, si $i < k$

$$\begin{aligned} \sum_{l=1}^{i} a_{i-1}^{(l-1)} p_{l-1}^{(0)} + p_i^{(0)} &= \sum_{l=0}^{i-1} a_{i-1}^{(l)} p_l^{(0)} + p_i^{(0)} \text{ par changement d'indice} \\ &= 0 \text{ si } i \neq k \text{ et } -p_k^{(0)} \text{ sinon d'après (II.15).} \end{aligned}$$

Quant aux colonnes suivantes, on trouve

$$\begin{aligned} \sum_{l=1}^{k} a_{i-1}^{(l-1)} p_{l-1}^{(j-1)} &= \sum_{l=0}^{k-1} a_{i-1}^{(l)} p_l^{(j-1)} \\ &= \sum_{l=0}^{i-1} a_{i-1}^{(l)} p_l^{(j-1)} + \sum_{l=i}^{k-1} a_{i-1}^{(l)} p_l^{(j-1)} \text{ si } i < k \\ &= p_{i-1}^{(j-2)} - p_i^{(j-1)} + \sum_{l=i}^{k-1} a_{i-1}^{(l)} p_l^{(j-1)} \text{ d'après (II.16)} \\ &= p_{i-1}^{(j-2)} - p_i^{(j-1)} + p_i^{(j-1)} \text{ car } a_{i-1}^{(i)} = 1 \text{ et } a_{i-1}^{(i+j)} = 0 \text{ si } j > 0 \\ &= p_{i-1}^{(j-2)}. \end{aligned}$$

Enfin, si $i = k$,

$$\sum_{l=1}^{k} a_{k-1}^{(l-1)} p_{l-1}^{(j-1)} = p_{k-1}^{(j-2)} - p_k^{(j-1)} \text{ si } j > 1.$$

Le produit matriciel $\boldsymbol{J}_k\boldsymbol{T}_k$ est donc la matrice de Hessenberg inférieure

$$\begin{pmatrix} 0 & p_0^{(0)} & 0 & \cdots & \cdots & 0 \\ 0 & p_1^{(0)} & p_1^{(1)} & 0 & \cdots & \vdots \\ \vdots & \vdots & \vdots & & & \vdots \\ 0 & p_{k-3}^{(0)} & p_{k-3}^{(1)} & \cdots & p_{k-3}^{(k-3)} & 0 \\ 0 & p_{k-2}^{(0)} & p_{k-2}^{(1)} & \cdots & p_{k-2}^{(k-3)} & p_{k-2}^{(k-2)} \\ -p_k^{(0)} & p_{k-1}^{(0)} - p_k^{(1)} & p_{k-1}^{(1)} - p_k^{(2)} & \cdots & p_{k-1}^{(k-3)} - p_k^{(k-2)} & p_{k-1}^{(k-2)} - p_k^{(k-1)} \end{pmatrix},$$

que l'on reconnaît sans difficulté comme étant l'expression du produit matriciel $\boldsymbol{T}_k\boldsymbol{F}_k$, ce qui achève la démonstration.

■

Nous obtiendrons de même que les matrices $\widetilde{\boldsymbol{J}}_k$ et $\widetilde{\boldsymbol{F}}_k$, où

$$\widetilde{\boldsymbol{F}}_k = \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & & & \vdots \\ -\widetilde{p}_k^{(0)} & -\widetilde{p}_k^{(1)} & \cdots & -\widetilde{p}_k^{(k-2)} & -\widetilde{p}_k^{(k-1)} \end{pmatrix}$$

est la matrice compagnon de $\widetilde{P}_k$, sont semblables avec $\widetilde{\boldsymbol{T}}_k\widetilde{\boldsymbol{F}}_k = \widetilde{\boldsymbol{J}}_k\widetilde{\boldsymbol{T}}_k$ par un raisonnement identique. On observe que les relations valables pour les polynômes orthogonaux s'étendent sans problème aux polynômes biorthogonaux.

Remarque 2.2

La relation (II.13) est en particulier valide pour les polynômes orthogonaux de dimension $d > 1$. La matrice de Hessenberg sera alors une matrice bande puisque la relation de récurrence ne comporte, pour ces polynômes, que $d+2$ termes.

2.2 Zéros des polynômes et relations matricielles

Nous allons voir ici que les matrices précédemment définies sont liées d'une certaine manière aux racines des polynômes biorthogonaux P_k et $\widetilde{P}_k$, ainsi qu'aux valeurs propres et vecteurs propres de certaines matrices.

Nous allons donc, dans cette sous-section, considérer les racines des polynômes biorthogonaux et voir s'il est possible d'étendre certains résultats connus sur les polynômes orthogonaux.

Théorème 2.1

Soit P_k le polynôme biorthogonal de degré k. Soit $\boldsymbol{J}_k$ la matrice de Hessenberg inférieure introduite en (II.11).

Alors

$$P_k(x) = |x\boldsymbol{I}_k - \boldsymbol{J}_k|$$

où $\boldsymbol{I}_k$ est la matrice identité de dimension k.
Par convention, on posera $P_0(x) = 1 = |x\boldsymbol{I}_0 - \boldsymbol{J}_0|$.

Démontrons tout d'abord le

Lemme 2.1
Soit $\boldsymbol{G}_k$ la matrice définie par

$$\boldsymbol{G}_k = \begin{pmatrix} a_{0,0} & -1 & 0 & \cdots & 0 \\ a_{1,0} & a_{1,1} & -1 & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ \vdots & \vdots & & \ddots & -1 \\ a_{k-1,0} & a_{k-1,1} & \cdots & \cdots & a_{k-1,k-1} \end{pmatrix}.$$

où les coefficients $(a_{i,j})_{0 \leqslant j \leqslant i \leqslant k-1}$ sont des complexes.
Alors,

$$|\boldsymbol{G}_k| = a_{k-1,0} + \sum_{i=1}^{k-1} a_{k-1,i} \, |\boldsymbol{G}_i| \, .$$

Preuve :

Si l'on développe par rapport à la dernière ligne, on trouve

$$|\boldsymbol{G}_k| = \sum_{i=0}^{k-1} (-1)^{k-1+i} a_{k-1,i} \left| \boldsymbol{G}_k^{(i)} \right|$$

où

$$\boldsymbol{G}_k^{(i)} = \begin{pmatrix} \boldsymbol{G}_i & 0 \\ \boldsymbol{G}' & \widetilde{\boldsymbol{G}}_k^{(i)} \end{pmatrix} \text{ si } i > 0$$

avec $\widetilde{\boldsymbol{G}}_k^{(i)}$ qui est une matrice triangulaire inférieure de dimension $k-i-1$ à diagonale -1 et $\boldsymbol{G}_k^{(0)} = (-1)^k a_{k,0}$.

Ainsi, $\left| \boldsymbol{G}_k^{(i)} \right| = (-1)^{k-i-1} \, |\boldsymbol{G}_i|$ et l'on obtient le résultat annoncé.

■

Nous pouvons maintenant donner la démonstration du Théorème 2.1.

Du Lemme 2.1 on déduit le résultat du Théorème 2.1. En effet, on raisonne par récurrence sur k :

- on vérifie que $|x\boldsymbol{I}_1 - \boldsymbol{J}_1| = x - a_0^{(0)} = P_1(x)$.
- On suppose que $|x\boldsymbol{I}_j - \boldsymbol{J}_j| = P_j(x)$ pour $j = 1, \ldots k-1$.
- Alors, d'après le Lemme 2.1,

$$\begin{aligned}
|x\boldsymbol{I}_k - \boldsymbol{J}_k| &= (x - a_{k-1}^{(k-1)})\,|x\boldsymbol{I}_{k-1} - \boldsymbol{J}_{k-1}| - \sum_{i=1}^{k-2} a_{k-1}^{(i)}\,|x\boldsymbol{I}_i - \boldsymbol{J}_i| - a_{k-1}^{(0)} \\
&= xP_{k-1}(x) - a_{k-1}^{(0)} - \sum_{i=1}^{k-1} a_{k-1}^{(i)} P_i(x) \text{ par hypothèse de récurrence} \\
&= xP_{k-1}(x) - \sum_{i=0}^{k-1} a_{k-1}^{(i)} P_i(x) \text{ car } P_0(x) = 1 \\
&= P_k(x).
\end{aligned}$$

Ceci achève la démonstration.

■

On remarque qu'il s'agit d'une généralisation du résultat connu sur les polynômes orthogonaux (Théorème 1.4 de la Première Partie). Il permet également d'obtenir une extension pour les polynômes orthogonaux de dimension $d > 1$.

Ce Théorème nous permet, bien entendu, de caractériser de façon matricielle les racines du polynôme P_k sous forme d'un Corollaire.

Corollaire 2.1.1

Si P_k est un polynôme biorthogonal et $\boldsymbol{J}_k$ désigne la matrice de Hessenberg inférieur introduite en (II.11), alors les zéros du polynôme P_k sont les valeurs propres de la matrice $\boldsymbol{J}_k$.

Preuve :

La démonstration est immédiate si l'on revient à la définition des valeurs propres d'une matrice et si l'on utilise le résultat du Théorème précédent.

■

On trouvera de même que les zéros du polynôme $\widetilde{P}_k$ sont les valeurs propres de la matrice $\widetilde{\boldsymbol{J}}_k$ et que

$$\widetilde{P}_k(x) = \left| x\boldsymbol{I}_k - \widetilde{\boldsymbol{J}}_k \right| .$$

Intéressons-nous de plus près aux zéros de ces polynômes biorthogonaux. Soit désormais $\boldsymbol{Z}_k$ la matrice $\mathrm{diag}(z_k^{(1)}, \ldots, z_k^{(k)})$ où les k scalaires $\left(z_k^{(i)}\right)_{1 \leqslant i \leqslant k}$ sont les k racines (comptées avec leur ordre de multiplicité) du polynôme P_k. Définissons de plus la matrice de Vandermonde

$$\boldsymbol{V}_k = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ z_k^{(1)} & z_k^{(2)} & \cdots & z_k^{(k)} \\ \vdots & \vdots & & \vdots \\ {z_k^{(1)}}^{k-1} & {z_k^{(2)}}^{k-1} & \cdots & {z_k^{(k)}}^{k-1} \end{pmatrix}$$

générée par ces zéros.

De même, pour la suite, on posera $\widetilde{\boldsymbol{Z}}_k = \mathrm{diag}(\widetilde{z}_k^{(1)}, \ldots, \widetilde{z}_k^{(k)})$ et

$$\widetilde{\boldsymbol{V}}_k = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ \widetilde{z}_k^{(1)} & \widetilde{z}_k^{(2)} & \cdots & \widetilde{z}_k^{(k)} \\ \vdots & \vdots & & \vdots \\ {\widetilde{z}_k^{(1)}}^{k-1} & {\widetilde{z}_k^{(2)}}^{k-1} & \cdots & {\widetilde{z}_k^{(k)}}^{k-1} \end{pmatrix},$$

où les complexes $\left(\widetilde{z}_k^{(i)}\right)_{1 \leqslant i \leqslant k}$ sont les k racines du polynôme $\widetilde{P}_k$, comptées, elles aussi, avec leur ordre de multiplicité.

En utilisant ces différentes matrices, on obtient la

Proposition 2.5

Si $\boldsymbol{V}_k$ est la matrice de Vandermonde générée par les zéros de P_k, si $\boldsymbol{F}_k$ est la matrice compagnon de P_k et $\boldsymbol{Z}_k$ la matrice diagonale composée des racines de P_k, alors les vecteurs $\left(1, z_k^{(i)}, \ldots, {z_k^{(i)}}^{k-1}\right)^{\mathrm{T}}$ pour $i = 1, 2, \ldots, k$ sont des vecteurs propres de $\boldsymbol{F}_k$ et de plus

$$\boldsymbol{F}_k \boldsymbol{V}_k = \boldsymbol{V}_k \boldsymbol{Z}_k.$$

Preuve :

L'égalité matricielle précédente est évidente pour les $k-1$ premières lignes. Pour la dernière, l'élément de la colonne j est

$$-\sum_{i=0}^{k-1} p_k^{(i)} {z_k^{(j)}}^i = {z_k^{(j)}}^k \text{ car } P_k(z_k^{(j)}) = 0$$

en ce qui concerne le produit matriciel $\boldsymbol{F}_k\boldsymbol{V}_k$, ce qui est clairement la dernière ligne du produit de $\boldsymbol{V}_k$ par $\boldsymbol{Z}_k$.

Ceci justifie l'assertion sur les vecteurs propres et achève la démonstration.

■

Ce résultat est une extension de l'égalité matricielle sur les polynômes orthogonaux qui a été rappelée dans la Première Partie en (I.3).

En outre, un cas particulier donne une égalité matricielle identique pour les polynômes orthogonaux de dimension $d > 1$.

Proposition 2.6

Si $\boldsymbol{J}_k$ est la matrice de Hessenberg inférieure introduite en (II.11), si $\boldsymbol{T}_k$ est la matrice des coefficients des polynômes biorthogonaux, si $\boldsymbol{Z}_k$ est la matrice diagonale composée des zéros de P_k et si $\boldsymbol{V}_k$ est la matrice de Vandermonde générée par ces racines, alors les matrices $\boldsymbol{J}_k$ et $\boldsymbol{Z}_k$ sont semblables et, en posant $\boldsymbol{Q}_k = \boldsymbol{T}_k\boldsymbol{V}_k$, on a

$$\boldsymbol{J}_k\boldsymbol{Q}_k = \boldsymbol{Q}_k\boldsymbol{Z}_k.$$

Preuve :

$\boldsymbol{J}_k\boldsymbol{T}_k\boldsymbol{V}_k = \boldsymbol{T}_k\boldsymbol{F}_k\boldsymbol{V}_k$ d'après la Proposition 2.4. Et $\boldsymbol{T}_k\boldsymbol{F}_k\boldsymbol{V}_k = \boldsymbol{T}_k\boldsymbol{V}_k\boldsymbol{Z}_k$ d'après la Proposition 2.5. En égalant les deux expressions, on obtient le résultat.

■

Cette dernière Proposition est une généralisation du Théorème 1.5 de la Première Partie portant sur les polynômes orthogonaux.

Toutes les relations fermées données ici nous permettent donc de généraliser celles exprimées dans [4] aux polynômes biorthogonaux et également aux polynômes orthogonaux de dimension $d > 1$.

D'autre part, il est bien évident que toutes les relations matricielles démontrées pour les polynômes biorthogonaux à gauche (c'est-à-dire les P_k) ont leur équivalent pour les polynômes biorthogonaux à droite (les $\widetilde{P}_k$). Elles ne prennent en compte qu'une seule famille de polynômes biorthogonaux.

2.3 Relations matricielles mixtes

Après avoir étudié les diverses relations matricielles qui pouvaient être tirées d'une unique famille de polynômes biorthogonaux, il serait intéressant de voir si l'on peut en trouver certaines qui feraient apparaître simultanément les deux familles $\{P_k\}_{k\geqslant 0}$ et $\left\{\widetilde{P}_k\right\}_{k\geqslant 0}$.

Pour cela, nous allons introduire ici deux nouvelles matrices, correspondant à l'écriture de (II.9) et de (II.10).

Posons

$$\boldsymbol{J'}_k = \begin{pmatrix} b_0^{(0)} & 1 & 0 & \dots & 0 \\ b_1^{(0)} & b_1^{(1)} & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ \vdots & \vdots & & \ddots & 1 \\ b_{k-1}^{(0)} & b_{k-1}^{(1)} & \dots & \dots & b_{k-1}^{(k-1)} \end{pmatrix} \tag{II.17}$$

et

$$\widetilde{\boldsymbol{J}}'_k = \begin{pmatrix} \widetilde{b}_0^{(0)} & 1 & 0 & \dots & 0 \\ \widetilde{b}_1^{(0)} & \widetilde{b}_1^{(1)} & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ \vdots & \vdots & & \ddots & 1 \\ \widetilde{b}_{k-1}^{(0)} & \widetilde{b}_{k-1}^{(1)} & \dots & \dots & \widetilde{b}_{k-1}^{(k-1)} \end{pmatrix}. \tag{II.18}$$

Ces matrices de Hessenberg inférieures définies, on obtient la

Proposition 2.7

Si les matrices $\boldsymbol{T}_k$ et $\widetilde{\boldsymbol{T}}_k$ représentent respectivement les coefficients des polynômes biorthogonaux à gauche et les coefficients des polynômes biorthogonaux à droite, si $\boldsymbol{F}_k$ est la matrice compagnon de P_k et si $\boldsymbol{J'}_k$ est la matrice de Hessenberg inférieure introduite en (II.17), alors ces matrices sont liées et vérifient la relation

$$\boldsymbol{J'}_k\boldsymbol{T}_k = \widetilde{\boldsymbol{T}}_k\boldsymbol{F}_k. \tag{II.19}$$

Preuve :

En remarquant, d'après (II.9), que

$$\begin{aligned} p_{k+1}^{(0)} &= -\sum_{j=0}^{k} b_k^{(j)} p_j^{(0)} \\ p_{k+1}^{(i)} &= \widetilde{p}_k^{(i-1)} - \sum_{j=i}^{k} b_k^{(j)} p_j^{(i)} \text{ si } 1 \leqslant i \leqslant k \\ p_{k+1}^{(k+1)} &= 1, \end{aligned}$$

et en raisonnant de façon analogue à la démonstration de la Proposition 2.4, on trouve le résultat.

■

De même, on trouvera la

Proposition 2.8

Si les matrices $\boldsymbol{T}_k$ et $\widetilde{\boldsymbol{T}}_k$ représentent respectivement les coefficients des polynômes biorthogonaux à gauche et les coefficients des polynômes biorthogonaux à droite, si $\widetilde{\boldsymbol{F}}_k$ est la matrice compagnon de $\widetilde{P}_k$ et si $\widetilde{\boldsymbol{J}'}_k$ est la matrice de Hessenberg inférieure introduite en (II.18), alors ces matrices sont liées par la relation

$$\widetilde{\boldsymbol{J}'}_k \widetilde{\boldsymbol{T}}_k = \boldsymbol{T}_k \widetilde{\boldsymbol{F}}_k. \tag{II.20}$$

Preuve :

La démonstration est identique à celle de la Proposition précédente.

■

Les deux Propositions précédentes nous permettent d'obtenir le

Théorème 2.2

Si la matrice $\widetilde{\boldsymbol{T}}_k$ représente les coefficients des polynômes biorthogonaux à droite, si $\boldsymbol{F}_k$ est la matrice compagnon de P_k, si $\boldsymbol{J'}_k$ est la matrice de Hessenberg inférieure introduite en (II.17), si $\widetilde{\boldsymbol{F}}_k$ est la matrice compagnon de $\widetilde{P}_k$ et si enfin $\widetilde{\boldsymbol{J}'}_k$ est la matrice de Hessenberg inférieure introduite en(II.18), alors les matrices $\boldsymbol{J'}_k \widetilde{\boldsymbol{J}'}_k$ et $\boldsymbol{F}_k \widetilde{\boldsymbol{F}}_k$ sont deux matrices semblables et elles vérifient

$$\boldsymbol{J'}_k \widetilde{\boldsymbol{J}'}_k \widetilde{\boldsymbol{T}}_k = \widetilde{\boldsymbol{T}}_k \boldsymbol{F}_k \widetilde{\boldsymbol{F}}_k.$$

Preuve :

Il suffit de multiplier (II.19) à droite par $\widetilde{\boldsymbol{F}}_k$ et d'utiliser l'égalité (II.20). Le résultat est alors immédiat.

■

De même on trouve le

Théorème 2.3

Si la matrice $\boldsymbol{T}_k$ représente les coefficients des polynômes biorthogonaux à gauche, si $\boldsymbol{F}_k$ est la matrice compagnon de P_k, si $\boldsymbol{J'}_k$ est la matrice de Hessenberg inférieure introduite en (II.17), si $\widetilde{\boldsymbol{F}}_k$ est la matrice compagnon de $\widetilde{P}_k$ et si enfin $\widetilde{\boldsymbol{J}}'_k$ est la matrice de Hessenberg inférieure introduite en (II.18), alors les matrices $\widetilde{\boldsymbol{J}}'_k\boldsymbol{J'}_k$ et $\widetilde{\boldsymbol{F}}_k\boldsymbol{F}_k$ sont deux matrices semblables et, à ce titre, elles vérifient

$$\widetilde{\boldsymbol{J}}'_k\boldsymbol{J'}_k\boldsymbol{T}_k = \boldsymbol{T}_k\widetilde{\boldsymbol{F}}_k\boldsymbol{F}_k.$$

Preuve :

Il suffit de multiplier à gauche (II.19) par $\widetilde{\boldsymbol{J}}'_k$ et d'utiliser l' égalité (II.20).

■

Enfin, sous une certaine condition, on trouve deux relations reliant toutes les matrices de coefficients.

Proposition 2.9

Si $\boldsymbol{J}_k$ et $\widetilde{\boldsymbol{J}}_k$ représentent respectivement les matrices de Hessenberg inférieures introduites en (II.11) et (II.12). Si $\boldsymbol{T}_k$ et $\widetilde{\boldsymbol{T}}_k$ sont respectivement les matrices des coefficients des polynômes biorthogonaux à gauche et à droite, si $\boldsymbol{J'}_k$ et $\widetilde{\boldsymbol{J}}'_k$ représentent respectivement les matrices de Hessenberg inférieures introduites en (II.17) et en (II.18) et si, de plus, $P_k(0) \neq 0$ et $\widetilde{P}_k(0) \neq 0$, alors ces matrices vérifient

$$\boldsymbol{J}_k\boldsymbol{J'}_k^{-1} = \widetilde{\boldsymbol{J}}'_k\widetilde{\boldsymbol{J}}_k^{-1} = \boldsymbol{T}_k\widetilde{\boldsymbol{T}}_k^{-1}.$$

Preuve :

Si $P_k(0) \neq 0$ alors, d'après la Proposition 2.7, $\boldsymbol{J'}_k$ sera inversible. En effet,

$$|\boldsymbol{T}_k| = |\widetilde{\boldsymbol{T}}_k| = 1,\ |\boldsymbol{F}_k| = (-1)^{k-1}P_k(0) \Longrightarrow |\boldsymbol{J}_k| = |\boldsymbol{F}_k| \neq 0.$$

De même, on trouve que si $\widetilde{P}_k(0) \neq 0$, alors $\widetilde{J}_k$ est inversible. On utilise les Propositions 2.4, 2.7 et 2.8. Le résultat est alors immédiat par passage à l'inverse.

■

Toutes les relations établies dans cette sous-section avaient leur équivalent pour les polynômes orthogonaux associés à une matrice de Hankel même si cela n'apparaît pas dans la Première Partie. En effet, certaines de ces relations sont alors confondues dans la mesure où, pour ces matrices, les polynômes biorthogonaux à gauche et à droite sont identiques (une matrice de Hankel est symétrique!). Par contre, pour les polynômes vectoriellement orthogonaux, ces relations n'étaient pas connues dans la mesure où l'on a considéré ici les deux familles de polynômes biorthogonaux simultanément.

3 Noyau reproduisant et identités de type Christoffel-Darboux

Lorsque l'on rencontre dans la littérature la notion de polynômes orthogonaux (et aussi de polynômes orthogonaux matriciels [77]), on trouve généralement également la notion d'identité de Christoffel-Darboux, ou de type Christoffel-Darboux. Associée au noyau reproduisant, cette notion permet d'obtenir des relations matricielles supplémentaires. Nous verrons, dans cette Section, que des relations du même type existent pour les polynômes biorthogonaux.

Il faudra alors tout d'abord définir la notion de noyau reproduisant dans le cadre des polynômes biorthogonaux et analyser ses propriétés principales, ce qui sera le but de la **première sous-section**.

Nous pourrons alors dans la **deuxième sous-section** déterminer les relations matricielles issues tout d'abord du noyau reproduisant.

Nous serons alors à même de donner des relations de type Christoffel-Darboux pour les polynômes biorthogonaux. Celles-ci seront regroupées dans la **troisième sous-section**.

3.1 Noyau reproduisant - Définition, propriétés

La notion de noyau reproduisant est liée à une certaine forme bilinéaire que nous allons définir dans un premier temps. Les propriétés du noyau reproduisant seront établies ensuite.

Notons P et Q les polynômes de degré respectif k_1 et k_2.

$$\begin{aligned} P(x) &= \sum_{i=0}^{k_1} p^{(i)} x^i \\ Q(x) &= \sum_{i=0}^{k_2} q^{(i)} x^i. \end{aligned}$$

On définit tout d'abord une forme bilinéaire dépendant de la matrice $\boldsymbol{L}_k$ de $\mathbb{C}[X] \times \mathbb{C}[X]$ dans $\mathbb{C}$ par

$$< P, Q >_{\boldsymbol{L}_k} = p^{\mathrm{T}} \boldsymbol{L}_k q \tag{II.21}$$

où $k \geqslant \max(k_1, k_2)$ et p (resp. q) désigne le vecteur de $\mathbb{C}^{k+1}$ dont les $k_1 + 1$ (resp. les $k_2 + 1$) composantes sont $p^{(i)}$, pour $i = 0, \ldots, k_1$ (resp. $q^{(i)}$, pour $i = 0, \ldots, k_2$), toutes les suivantes étant nulles.

Rappel 3.1

Les coefficients que l'on note d_i sont les éléments diagonaux qui apparaissent dans la matrice diagonale définie à la Proposition 2.3 par le produit $\boldsymbol{D}_k = \widetilde{\boldsymbol{T}}_k \boldsymbol{L}_k \boldsymbol{T}_k^{\mathrm{T}}$.

Ainsi, on énonce la

Propriété 3.1

Soient $\widetilde{P}_i$ le polynôme biorthogonal à droite de degré i et P_j le polynôme biorthogonal à gauche de degré j.

Alors, la forme bilinéaire définie précédemment vérifie

$$< \widetilde{P}_i, P_j >_{\boldsymbol{L}_k} = \delta_{i,j} d_i$$

où $\delta_{i,j}$ est le symbole de Kronecker ($\delta_{i,j} = 1$ si $i = j$ et 0 sinon).

Preuve :

Il suffit de constater que cette expression n'est autre que le coefficient de la ligne i et de la colonne j de la matrice $\boldsymbol{D}_k$ du Rappel 3.1. C'est en fait une conséquence directe de la Proposition 2.3.

■

Cette forme bilinéaire étant définie, il est possible maintenant d'introduire la notion de noyau reproduisant.

Définition 3.1 - *Noyau reproduisant (Bultheel et al. [24])*

Si les polynômes P_k et $\widetilde{P}_k$ sont les polynômes biorthogonaux respectivement à gauche et à droite de $\boldsymbol{L}_k$, alors, le polynôme à deux variables défini par

$$\Omega_k(x, \phi) = \sum_{i=0}^{k} P_i(\phi) d_i^{-1} \widetilde{P}_i(x),$$

où $d_i =< \widetilde{P}_i, P_i >_{\boldsymbol{L}_k}$, sera appelé noyau reproduisant relatif à la matrice $\boldsymbol{L}_k$.

Les quantités d_i^{-1} ont bien un sens car la matrice $\boldsymbol{L}_k$ est supposée fortement régulière. Or, d'après la Proposition 2.3, $|\boldsymbol{D}_k| = |\widetilde{\boldsymbol{T}}_k| \, |\boldsymbol{L}_k| \, |\boldsymbol{T}_k| \neq 0$ car tous les polynômes biorthogonaux sont supposés exister. Ainsi, aucun élément de la matrice diagonale $\boldsymbol{D}_k$ ne peut être nul.

Il est à noter qu'une autre définition du noyau reproduisant (cette fois-ci matricielle) relatif à deux suites de polynômes a été introduite par Van Iseghem

et al. dans [77].

Voyons les propriétés essentielles du noyau reproduisant introduit à la Définition 3.1. Celles-ci sont associées également à la forme bilinéaire définie précédemment.

Propriété 3.2

Soit Q_k un polynôme de degré au plus k. Alors le noyau reproduisant et la forme bilinéaire définis plus haut vérifient

$$< Q_k(\phi), \Omega_k(x,\phi) >_{\boldsymbol{L}_k} = Q_k(x),$$

lorsque la forme bilinéaire agit sur ϕ et que x est un paramètre. Ceci justifie l'appellation de noyau reproduisant.

Preuve :

Les polynômes $\left\{\widetilde{P}_i\right\}_{0\leqslant i\leqslant k}$ étant de degré i exactement, ils forment une base de $\mathcal{P}_k$.
À ce titre on peut écrire

$$Q_k(\phi) = \sum_{j=0}^{k} \widetilde{q}_k^{(j)} \widetilde{P}_j(\phi).$$

Ainsi,

$$\begin{aligned}
< Q_k(\phi), \Omega_k(x,\phi) >_{\boldsymbol{L}_k} &= < \sum_{j=0}^{k} \widetilde{q}_k^{(j)} \widetilde{P}_j(\phi), \sum_{i=0}^{k} P_i(\phi) d_i^{-1} \widetilde{P}_i(x) >_{\boldsymbol{L}_k} \\
&= \sum_{j=0}^{k} \widetilde{q}_k^{(j)} \sum_{i=0}^{k} < \widetilde{P}_j(\phi), P_i(\phi) d_i^{-1} \widetilde{P}_i(x) >_{\boldsymbol{L}_k} \\
&= \sum_{j=0}^{k} \widetilde{q}_k^{(j)} \sum_{i=0}^{k} d_i^{-1} \widetilde{P}_i(x) < \widetilde{P}_j(\phi), P_i(\phi) >_{\boldsymbol{L}_k} \\
&= \sum_{j=0}^{k} \widetilde{q}_k^{(j)} \sum_{i=0}^{k} d_i^{-1} \widetilde{P}_i(x) \delta_{i,j} d_i \\
&= \sum_{j=0}^{k} \widetilde{q}_k^{(j)} \widetilde{P}_j(x) = Q_k(x).
\end{aligned}$$

■

De même, on trouvera la

Propriété 3.3

Soit Q_k un polynôme de degré au plus k. Alors le noyau reproduisant et la forme bilinéaire définis plus haut vérifient

$$< \Omega_k(x,\phi), Q_k(x) >_{L_k} = Q_k(\phi)$$

lorsque la forme bilinéaire agit sur x et que ϕ est un paramètre.

Preuve :

Il suffit d'écrire

$$Q_k(x) = \sum_{j=0}^{k} q_k^{(j)} P_j(x)$$

et de raisonner de façon strictement analogue à la Propriété précédente.

■

La notion de noyau reproduisant existant pour les polynômes orthogonaux se généralise donc très bien aux polynômes biorthogonaux.

3.2 Relations matricielles et noyau reproduisant

Après avoir défini la notion de noyau reproduisant, nous allons établir des relations matricielles qui impliquent ce dernier. Nous démontrerons notamment des relations entre les zéros des polynômes biorthogonaux et la valeur du noyau reproduisant en ces racines.

Définissons ici trois nouvelles matrices issues du noyau reproduisant qui nous seront utiles par la suite.

Posons

$$\boldsymbol{W}_k = \left(\Omega_k(z_k^{(j)}, z_k^{(i)})\right)_{1\leqslant i,j\leqslant k}, \tag{II.22}$$

$$\widetilde{\boldsymbol{W}}_k = \left(\Omega_k(\widetilde{z}_k^{(j)}, \widetilde{z}_k^{(i)})\right)_{1\leqslant i,j\leqslant k}, \tag{II.23}$$

$$\boldsymbol{W'}_k = \left(\Omega_k(\widetilde{z}_k^{(j)}, z_k^{(i)})\right)_{1\leqslant i,j\leqslant k}. \tag{II.24}$$

Les coefficients de ces matrices sont donc les valeurs du noyau reproduisant pris en des points particuliers : les racines du polynôme biorthogonal de degré k à gauche et les zéros du polynôme biorthogonal de degré k à droite.

Remarque 3.1

Dans le cas des polynômes orthogonaux (généralement associé aux matrices de Hankel), ces trois matrices sont les mêmes puisque la matrice des formes linéaires considérées est symétrique. De plus elles sont diagonales (ce résultat est dû à l'identité de Christoffel-Darboux pour les polynômes orthogonaux). Ce caractère n'est malheureusement plus valable dans le cadre des polynômes biorthogonaux.

Ainsi définies, ces matrices nous permettent d'énoncer la

Proposition 3.1

Soit $\boldsymbol{L}_k$ une matrice de dimension k fortement régulière, $\boldsymbol{V}_k$ la matrice de Vandermonde générée par les racines de P_k. Soit $\boldsymbol{W}_k$ la matrice définie en (II.22).

Alors ces matrices vérifient

$$\boldsymbol{W}_k = \boldsymbol{V}_k^{\mathrm{T}} \boldsymbol{L}_k^{-1} \boldsymbol{V}_k.$$

Preuve :

Remarquons tout d'abord que l'élément de la ligne i et de la colonne j du produit matriciel $\widetilde{\boldsymbol{T}}_k \boldsymbol{V}_k$ est $\widetilde{P}_{i-1}(z_k^{(j)})$. On trouve alors que l'élément de la ligne i et de la colonne j de $\boldsymbol{Q}_k^{\mathrm{T}} \boldsymbol{D}_k^{-1} \widetilde{\boldsymbol{T}}_k \boldsymbol{V}_k$ est

$$\begin{aligned}
\sum_{l=1}^{k} P_{l-1}(z_k^{(i)}) d_{l-1}^{-1} \widetilde{P}_{l-1}(z_k^{(j)}) &= \sum_{l=0}^{k-1} P_l(z_k^{(i)}) d_l^{-1} \widetilde{P}_l(z_k^{(j)}) \\
&= \sum_{l=0}^{k} P_l(z_k^{(i)}) d_l^{-1} \widetilde{P}_l(z_k^{(j)}) \text{ car } P_k(z_k^{(i)}) = 0 \\
&= \Omega_k(z_k^{(j)}, z_k^{(i)}).
\end{aligned}$$

Ceci achève la démonstration.

■

De même, en raisonnant sur les polynômes biorthogonaux à droite de $\boldsymbol{L}_k$, on trouvera la

Proposition 3.2

Soit $\boldsymbol{L}_k$ une matrice de dimension k fortement régulière, $\widetilde{\boldsymbol{W}}_k$ la matrice définie en (II.23). Soit $\widetilde{\boldsymbol{V}}_k$ la matrice de Vandermonde générée par les zéros de $\widetilde{P}_k$.

Alors,

$$\widetilde{\boldsymbol{W}}_k = \widetilde{\boldsymbol{V}}_k^{\mathrm{T}} \boldsymbol{L}_k^{-1} \widetilde{\boldsymbol{V}}_k.$$

Preuve :

L'élément de la ligne i et de la colonne j de la matrice $\boldsymbol{T}_k\widetilde{\boldsymbol{V}}_k$ est $P_{i-1}(\widetilde{z}_k^{(j)})$. Alors, l'élément de la ligne i et de la colonne j de $\widetilde{\boldsymbol{V}}_k\boldsymbol{T}_k^{\mathrm{T}}\boldsymbol{D}_k^{-1}\widetilde{\boldsymbol{Q}}_k$ est

$$\begin{aligned}\sum_{l=1}^{k} P_{l-1}(\widetilde{z}_k^{(i)})d_{l-1}^{-1}\widetilde{P}_{l-1}(\widetilde{z}_k^{(j)}) &= \sum_{l=0}^{k-1} P_l(\widetilde{z}_k^{(i)})d_l^{-1}\widetilde{P}_l(\widetilde{z}_k^{(j)}) \\ &= \sum_{l=0}^{k} P_l(\widetilde{z}_k^{(i)})d_l^{-1}\widetilde{P}_l(\widetilde{z}_k^{(j)}) \text{ car } \widetilde{P}_k(\widetilde{z}_k^{(j)}) = 0 \\ &= \Omega_k(\widetilde{z}_k^{(j)}, \widetilde{z}_k^{(i)}).\end{aligned}$$

Ceci achève la démonstration.

■

Maintenant, après avoir considéré séparément les zéros des polynômes des familles $\{P_k\}_{k>0}$ et $\left\{\widetilde{P}_k\right\}_{k>0}$, nous allons pouvoir mettre en évidence des relations où les racines des polynômes des deux familles sont liées.

Ainsi, on obtient le

Théorème 3.1

Si la matrice $\widetilde{\boldsymbol{T}}_k$ est la matrice des coefficients des polynômes biorthogonaux à droite, si $\boldsymbol{D}_k$ est la matrice du Rappel 3.1, si $\boldsymbol{Q}_k$ est la matrice définie à la Proposition 2.6 et si enfin $\boldsymbol{W'}_k$ est la matrice définie en (II.24), alors, en posant $\widetilde{\boldsymbol{Q}}_k = \widetilde{\boldsymbol{T}}_k\widetilde{\boldsymbol{V}}_k$, on trouve

$$\boldsymbol{W'}_k = \boldsymbol{Q}_k^{\mathrm{T}}\boldsymbol{D}_k^{-1}\widetilde{\boldsymbol{Q}}_k.$$

Preuve :

Remarquons tout d'abord que l'élément de la ligne i et de la colonne j de $\boldsymbol{Q}_k$ est $P_{i-1}(z_k^{(j)})$. De même, l'élément de la ligne i et de la colonne j de $\widetilde{\boldsymbol{Q}}_k$ est $\widetilde{P}_{i-1}(\widetilde{z}_k^{(j)})$ (c'est clair). On rappelle que la matrice $\boldsymbol{D}_k$ est inversible si $\boldsymbol{L}_k$ est fortement régulière. L'élément de la ligne i et de la colonne j du produit matriciel de $\boldsymbol{D}_k^{-1}$ par $\widetilde{\boldsymbol{Q}}_k$ sera alors $d_{i-1}^{-1}\widetilde{P}_{i-1}(\widetilde{z}_k^{(j)})$. Le dernier produit matriciel

nous donnera que l'élément de la ligne i et de la colonne j de $\boldsymbol{Q}_k^{\mathrm{T}}\boldsymbol{D}_k^{-1}\widetilde{\boldsymbol{Q}}_k$ est

$$\begin{aligned}\sum_{l=1}^{k} P_{l-1}(z_k^{(i)})d_{l-1}^{-1}\widetilde{P}_{l-1}(\widetilde{z}_k^{(j)}) &= \sum_{l=0}^{k-1} P_l(z_k^{(i)})d_l^{-1}\widetilde{P}_l(\widetilde{z}_k^{(j)}) \\ &= \sum_{l=0}^{k} P_l(z_k^{(i)})d_l^{-1}\widetilde{P}_l(\widetilde{z}_k^{(j)}) \text{ car } P_k(z_k^{(i)}) = 0 \\ &= \Omega_k(\widetilde{z}_k^{(j)}, z_k^{(i)}),\end{aligned}$$

ce qui achève la démonstration.

■

Ce Théorème est une extension de la Proposition 1.2 sur les polynômes orthogonaux. De ce Théorème, on déduit immédiatement le

Corollaire 3.1.1

Soit $\boldsymbol{L}_k$ une matrice de dimension k fortement régulière, $\boldsymbol{V}_k$ et $\widetilde{\boldsymbol{V}}_k$ les matrices de Vandermonde générées par les zéros respectifs des polynômes biorthogonaux à gauche et à droite par rapport à $\boldsymbol{L}_k$. Soit $\boldsymbol{W'}_k$ la matrice définie en (II.24).

Alors ces matrices vérifient

$$\boldsymbol{W'}_k = \boldsymbol{V}_k^{\mathrm{T}}\boldsymbol{L}_k^{-1}\widetilde{\boldsymbol{V}}_k.$$

Preuve :

De la Proposition 2.3, on déduit que

$$\boldsymbol{L}_k^{-1} = \boldsymbol{T}_k^{\mathrm{T}}\boldsymbol{D}_k^{-1}\widetilde{\boldsymbol{T}}_k. \tag{II.25}$$

En multipliant à gauche par $\boldsymbol{V}_k^{\mathrm{T}}$ et à droite par $\widetilde{\boldsymbol{V}}_k$, on trouve le résultat.

■

En particulier, dans le cas où la matrice $\boldsymbol{L}_k$ est symétrique, on déduit le

Corollaire 3.1.2

Si $\boldsymbol{L}_k$ est une matrice symétrique fortement régulière, si $\boldsymbol{V}_k$ est la matrice de Vandermonde générée par les racines de P_k et si les trois matrices $\boldsymbol{W}_k$, $\boldsymbol{W'}_k$ et $\widetilde{\boldsymbol{W}}_k$ sont celles définies respectivement en (II.22), (II.23) et (II.24), alors

$$\boldsymbol{W}_k = \widetilde{\boldsymbol{W}}_k = \boldsymbol{W'}_k = \boldsymbol{V}_k^{\mathrm{T}}\boldsymbol{L}_k^{-1}\boldsymbol{V}_k.$$

Preuve :

Ceci est uniquement dû au fait qu'une matrice symétrique a ses polynômes biorthogonaux à gauche et à droite identiques. (puisque les formes linéaires associées $\mathcal{L}_i$ et $\mathcal{L}_i^{\mathrm{T}}$ sont les mêmes).

■

Ce Corollaire généralise à nouveau les résultats connus sur les polynômes orthogonaux [4] et repris au Théorème 1.6 de la Première Partie. En particulier, l'expression du Corollaire précédent est la même que celle obtenue pour les polynômes orthogonaux et peut donc être étendue aux matrices symétriques (dont les matrices de Hankel n'en sont qu'une partie). Toutefois, la forme diagonale de $\boldsymbol{W}_k$ n'est généralement assurée que pour les matrices de Hankel.

Nous allons voir maintenant que, sous certaines hypothèses, les matrices $\boldsymbol{W}_k$, $\boldsymbol{W'}_k$ et $\widetilde{\boldsymbol{W}}_k$ sont liées.

Proposition 3.3

Si les racines du polynôme biorthogonal P_k sont toutes distinctes et qu'il en est de même pour celles du polynôme biorthogonal $\widetilde{P}_k$, alors les matrices $\boldsymbol{W}_k$, $\boldsymbol{W'}_k$ et $\widetilde{\boldsymbol{W}}_k$ définies respectivement en (II.22), (II.24) et (II.23) vérifient

$$\boldsymbol{W'}_k^{-\mathrm{T}}\widetilde{\boldsymbol{W}}_k^{\mathrm{T}} = \boldsymbol{W}_k^{-1}\boldsymbol{W'}_k = \boldsymbol{V}_k^{-1}\widetilde{\boldsymbol{V}}_k.$$

Preuve :

On utilise le Corollaire 3.1.1 puis les Propositions 3.1 et 3.2. De plus, la matrice $\boldsymbol{W'}_k$ est inversible si et seulement si les matrices $\boldsymbol{V}_k$, $\widetilde{\boldsymbol{V}}_k$ le sont (puisque $\boldsymbol{L}_k^{-1}$ est supposée l'être). Or, ces deux matrices étant de Vandermonde, elles sont inversibles si et seulement si les vecteurs les générant ont des composantes deux à deux distinctes, c'est-à-dire si les zéros de P_k ainsi que ceux de $\widetilde{P}_k$ sont deux à deux distincts.

■

Enfin, en utilisant les Propositions précédentes, on obtient le

Théorème 3.2

Les matrices $\boldsymbol{L}_k$ ainsi que les matrices $\boldsymbol{W}_k^{\mathrm{T}}$ et $\widetilde{\boldsymbol{W}}_k^{\mathrm{T}}$ définies respectivement en (II.22) et (II.23) sont congruentes et vérifient

$$\boldsymbol{L}_k = \widetilde{\boldsymbol{X}}_k \boldsymbol{W}_k^{\mathrm{T}} \widetilde{\boldsymbol{X}}_k^{\mathrm{T}}$$

et

$$\boldsymbol{L}_k = \boldsymbol{X}_k \widetilde{\boldsymbol{W}}_k^{\mathrm{T}} \boldsymbol{X}_k^{\mathrm{T}}$$

où $\widetilde{\boldsymbol{X}}_k = \widetilde{\boldsymbol{V}}_k \boldsymbol{W}'^{-1}_k$ *et* $\boldsymbol{X}_k = \boldsymbol{V}_k \boldsymbol{W}'^{-\mathrm{T}}_k$.

Preuve :

On utilise le Corollaire 3.1.1 et la Proposition 3.1, ainsi la première égalité en découle. D'autre part, en utilisant à nouveau le Corollaire 3.1.1 et la Proposition 3.2, la deuxième égalité est évidente.

■

Ces relations étaient pour la plupart confondues dans le cas des polynômes orthogonaux. Ces derniers sont en effet généralement associés aux matrices de Hankel, qui sont symétriques. Alors, il vient trivialement $\boldsymbol{L}_k = \boldsymbol{L}_k{}^{\mathrm{T}}$, ce qui implique $P_k = \widetilde{P}_k$, $\boldsymbol{V}_k = \widetilde{\boldsymbol{V}}_k$ et $\boldsymbol{W}_k = \widetilde{\boldsymbol{W}}_k = \boldsymbol{W}'_k$ ...

D'autre part, ces résultats peuvent s'étendre aux polynômes orthogonaux de dimension $d > 1$ (qui seront les P_k). Seulement, il faudra considérer la matrice de Hankel généralisée relative à ces polynômes ainsi que les polynômes biorthogonaux à droite (qui seront les $\widetilde{P}_k$) qui lui sont associés. La matrice $\boldsymbol{J}_k$ aura une forme particulière (ce sera une matrice bande puisque les polynômes orthogonaux de dimension $d > 1$ ne sont liés que par une relation de récurrence à $d+2$ termes).

3.3 Identités de type Christoffel-Darboux

Pour les polynômes orthogonaux, une identité de Christoffel-Darboux bien connue a été énoncée (voir par exemple [4]) et rappelée dans la Première Partie. En outre, pour les polynômes orthogonaux matriciels, une identité de type Christoffel-Darboux a pu être démontrée [77]. Elle peut mener à une identité de type Christoffel-Darboux pour les polynômes vectoriellement orthogonaux mais la forme bilinéaire utilisée diffère de celle utilisée ici. Nous nous intéresserons donc a une autre expression de cette identité.

Nous allons donc étudier ce que peut devenir l'identité de Christoffel-Darboux dans le cas des polynômes biorthogonaux et quelles peuvent être alors les diverses façons de l'écrire, en fonction des relations matricielles déjà obtenues à la Section 2.

Afin de trouver une identité de ce type, nous allons rappeler tout d'abord une certaine conception de cette notion qui permet l'expression du Théorème 1.3 de la Première Partie.

En effet, remarquons d'abord que l'identité de Christoffel-Darboux dans le cas des polynômes orthogonaux consiste en une écriture différente de l'expression du polynôme à deux variables $(x-\phi)\Omega_k(x,\phi)$. Ainsi, on écrit $x\Omega_k(x,\phi)$ puis on remplace les occurrences de $xP_i(x)$ par son expression obtenue en utilisant la relation de récurrence à trois termes du Théorème 1.2. On procède de même pour $\phi\Omega_k(x,\phi)$ en remplaçant bien sûr les occurrences de $\phi P_i(\phi)$. Alors, dans le cas des polynômes orthogonaux, certaines quantités se simplifient et l'on obtient l'identité de Christoffel-Darboux du Théorème 1.3 de la Première Partie.

Pour les polynômes biorthogonaux, on procédera de même et les relations matricielles obtenues à la Section 2 nous seront très utiles.

On obtient ainsi le

Théorème 3.3

Soit $\boldsymbol{D}_k$ la matrice diagonale du Rappel 3.1. Soient $\boldsymbol{J}_k$ et $\widetilde{\boldsymbol{J}}_k$ les matrices de Hessenberg introduites respectivement en (II.11) et en (II.12). Alors, une identité de type Christoffel-Darboux pour les polynômes biorthogonaux peut s'écrire

$$\begin{aligned}(x-\phi)\Omega_k(x,\phi) &= \begin{pmatrix} P_0(\phi) \\ \vdots \\ P_k(\phi) \end{pmatrix}^{\mathrm{T}} \left(\boldsymbol{D}_{k+1}^{-1}\widetilde{\boldsymbol{J}}_{k+1} - \boldsymbol{J}_{k+1}^{\mathrm{T}}\boldsymbol{D}_{k+1}^{-1}\right) \begin{pmatrix} \widetilde{P}_0(x) \\ \vdots \\ \widetilde{P}_k(x) \end{pmatrix} \\ &+ \frac{1}{d_k}\left(P_k(\phi)\widetilde{P}_{k+1}(x) - \widetilde{P}_k(x)P_{k+1}(\phi)\right).\end{aligned}$$

Preuve :

Cherchons tout d'abord une expression de $x\Omega_k(x,\phi)$. En remplaçant l'expression de $x\widetilde{P}_i(x)$ par celle de (II.8), on trouve

$$x\Omega_k(x,\phi) = \sum_{i=0}^{k} P_i(\phi)d_i^{-1}\left(\sum_{j=0}^{i+1}\widetilde{a}_i^{(j)}\widetilde{P}_j(x)\right)$$

si l'on pose $\widetilde{a}_i^{(i+1)} = 1$. On reconnaît là l'expression de

$$\begin{pmatrix} d_0^{-1}P_0(\phi) \\ \vdots \\ d_k^{-1}P_k(\phi) \end{pmatrix}^{\mathrm{T}} \begin{pmatrix} \boxed{\widetilde{\boldsymbol{J}}_{k+1}} & \begin{matrix} 0 \\ \vdots \\ 1 \end{matrix} \end{pmatrix} \begin{pmatrix} \widetilde{P}_0(x) \\ \vdots \\ \widetilde{P}_{k+1}(x) \end{pmatrix}.$$

D'où

$$x\Omega_k(x,\phi) = \begin{pmatrix} P_0(\phi) \\ \vdots \\ P_k(\phi) \end{pmatrix}^{\mathrm{T}} \boldsymbol{D}_{k+1}^{-1}\widetilde{\boldsymbol{J}}_{k+1} \begin{pmatrix} \widetilde{P}_0(x) \\ \vdots \\ \widetilde{P}_k(x) \end{pmatrix} + d_k^{-1}P_k(\phi)\widetilde{P}_{k+1}(x). \qquad \text{(II.26)}$$

En raisonnant de même avec $\phi\Omega_k(x,\phi)$, on trouve

$$\begin{aligned} \phi\Omega_k(x,\phi) &= \begin{pmatrix} \widetilde{P}_0(x) \\ \vdots \\ \widetilde{P}_k(x) \end{pmatrix}^{\mathrm{T}} \boldsymbol{D}_{k+1}^{-1}\boldsymbol{J}_{k+1} \begin{pmatrix} P_0(\phi) \\ \vdots \\ P_k(\phi) \end{pmatrix} + d_k^{-1}\widetilde{P}_k(x)P_{k+1}(\phi) \\ &= \begin{pmatrix} P_0(\phi) \\ \vdots \\ P_k(\phi) \end{pmatrix}^{\mathrm{T}} \boldsymbol{J}_{k+1}^{\mathrm{T}}\boldsymbol{D}_{k+1}^{-1} \begin{pmatrix} \widetilde{P}_0(x) \\ \vdots \\ \widetilde{P}_k(x) \end{pmatrix} + d_k^{-1}\widetilde{P}_k(x)P_{k+1}(\phi). \end{aligned}$$

Ceci achève la démonstration.

■

On trouve ainsi un *terme matriciel correcteur* à l'identité classique de Christoffel-Darboux pour les polynômes orthogonaux.

En effet, dans le cas où $\boldsymbol{L}_k$ est une matrice de Hankel, les polynômes P_k et $\widetilde{P}_k$ sont des polynômes orthogonaux formels identiques. Les matrices $\boldsymbol{J}_k$ et $\widetilde{\boldsymbol{J}}_k$ sont égales (puisque la matrice $\boldsymbol{L}_k$ est alors symétrique). Ainsi, en remarquant que les quantités $c(x^kP_k)$ introduites dans la Première Partie valent $\mathcal{L}_k(P_k)$, on vérifie que ce terme correcteur (qui est alors une matrice de Jacobi) est nul.

En utilisant la relation (II.13), on trouve également le

Corollaire 3.3.1

Soit $\boldsymbol{L}_k$ la matrice des formes linéaires associées aux polynômes biorthogonaux à gauche P_k et soient $\boldsymbol{F}_k$ et $\widetilde{\boldsymbol{F}}_k$ les matrices compagnons respectivement de P_k et $\widetilde{P}_k$.

Alors une identité de type Christoffel-Darboux peut s'écrire

$$\begin{aligned} (x-\phi)\Omega_k(x,\phi) &= \begin{pmatrix} 1 \\ \vdots \\ \phi^k \end{pmatrix}^{\mathrm{T}} \left(\boldsymbol{L}_{k+1}^{-1}\widetilde{\boldsymbol{F}}_{k+1} - \boldsymbol{F}_{k+1}^{\mathrm{T}}\boldsymbol{L}_{k+1}^{-1}\right) \begin{pmatrix} 1 \\ \vdots \\ x^k \end{pmatrix} \\ &\quad + \frac{1}{d_k}\left(P_k(\phi)\widetilde{P}_{k+1}(x) - \widetilde{P}_k(x)P_{k+1}(\phi)\right). \end{aligned}$$

Preuve :

Il suffit de réécrire l'expression de $x\Omega_k(x,\phi)$ à partir de (II.26) en remarquant que

$$\begin{pmatrix} P_0(\phi) \\ \vdots \\ P_k(\phi) \end{pmatrix} = \boldsymbol{T}_{k+1} \begin{pmatrix} 1 \\ \vdots \\ \phi^k \end{pmatrix}.$$

On remarque de même que

$$\begin{pmatrix} \widetilde{P}_0(x) \\ \vdots \\ \widetilde{P}_k(x) \end{pmatrix} = \widetilde{\boldsymbol{T}}_{k+1} \begin{pmatrix} 1 \\ \vdots \\ x^k \end{pmatrix}.$$

Ainsi, (II.26) devient

$$\begin{pmatrix} 1 \\ \vdots \\ \phi^k \end{pmatrix}^{\mathrm{T}} \boldsymbol{T}_{k+1}^{\mathrm{T}} \boldsymbol{D}_{k+1}^{-1} \widetilde{\boldsymbol{J}}_{k+1} \widetilde{\boldsymbol{T}}_{k+1} \begin{pmatrix} 1 \\ \vdots \\ x^k \end{pmatrix}.$$

Enfin, en utilisant (II.13), la Proposition 2.3 et en raisonnant de façon analogue pour $\phi\Omega_k(x,\phi)$, on trouve le résultat.

■

Une dernière expression peut être obtenue pour l'identité de type Christoffel-Darboux

Corollaire 3.3.2

Soit $\boldsymbol{L}_k$ la matrice des formes linéaires associées aux polynômes biorthogonaux à gauche P_k et soient $\boldsymbol{F}_k$ et $\widetilde{\boldsymbol{F}}_k$ les matrices compagnons respectivement de P_k et de $\widetilde{P}_k$.

Alors une identité de type Christoffel-Darboux peut s'écrire

$$\begin{aligned}(x-\phi)\Omega_k(x,\phi) &= \begin{pmatrix} 1 \\ \vdots \\ \phi^k \end{pmatrix}^{\mathrm{T}} \left(\boldsymbol{L}_{k+1}^{-1} \boldsymbol{F}_{k+1} - \widetilde{\boldsymbol{F}}_{k+1}^{\mathrm{T}} \boldsymbol{L}_{k+1}^{-1} \right) \begin{pmatrix} 1 \\ \vdots \\ x^k \end{pmatrix} \\ &+ \frac{1}{d_k} \left(P_k(\phi) P_{k+1}(x) - \widetilde{P}_k(x) \widetilde{P}_{k+1}(\phi) \right).\end{aligned}$$

Preuve :

Pour cela, nous allons écrire différemment $x\Omega_k(x,\phi)$. Nous obtenons

$$x\Omega_k(x,\phi) = \sum_{i=0}^{k} P_i(\phi) d_i^{-1} \left(\sum_{j=0}^{i+1} b_i^{(j)} P_j(x) \right)$$

si l'on pose $b_i^{(i+1)} = 1$. On reconnaît là l'expression de

$$\begin{pmatrix} d_0^{-1}P_0(\phi) \\ \vdots \\ d_k^{-1}P_k(\phi) \end{pmatrix}^{\mathrm{T}} \begin{pmatrix} \boxed{\boldsymbol{J'}_{k+1}} & \begin{matrix} 0 \\ \vdots \\ 1 \end{matrix} \end{pmatrix} \begin{pmatrix} P_0(x) \\ \vdots \\ P_{k+1}(x) \end{pmatrix}.$$

D'où

$$x\Omega_k(x,\phi) = \begin{pmatrix} P_0(\phi) \\ \vdots \\ P_k(\phi) \end{pmatrix}^{\mathrm{T}} \boldsymbol{D}_{k+1}^{-1} \boldsymbol{J'}_{k+1} \begin{pmatrix} P_0(x) \\ \vdots \\ P_k(x) \end{pmatrix} + d_k^{-1} P_k(\phi) P_{k+1}(x).$$

On trouve de même

$$\phi\Omega_k(x,\phi) = \begin{pmatrix} \widetilde{P}_0(x) \\ \vdots \\ \widetilde{P}_k(x) \end{pmatrix}^{\mathrm{T}} \boldsymbol{D}_{k+1}^{-1} \widetilde{\boldsymbol{J}}'_{k+1} \begin{pmatrix} \widetilde{P}_0(\phi) \\ \vdots \\ \widetilde{P}_k(\phi) \end{pmatrix} + d_k^{-1} \widetilde{P}_k(\phi) \widetilde{P}_{k+1}(x).$$

En utilisant enfin les égalités (II.19) et (II.20), on achève la démonstration.

■

Nous venons ainsi d'obtenir trois identités de type Christoffel-Darboux dont l'écriture est différente, selon que l'on considère une seule suite de polynômes biorthogonaux ou les deux suites issues d'une même matrice. Ces trois identités n'en deviennent bien entendu qu'une seule dans le cas où la matrice considérée est symétrique.

Pour les deux dernières identités, on remarque une forte analogie entre les deux expressions.

4 Un générateur de projecteurs orthogonaux

Dans cette Section, nous allons généraliser des résultats de [4] et [41] concernant les matrices résolvantes.

À partir de certaines relations matricielles établies plus haut, nous verrons qu'il apparaît certains projecteurs orthogonaux matriciels que nous nous proposons de caractériser.

La matrice $\boldsymbol{J}_k$ désigne toujours la matrice de Hessenberg inférieure introduite en (II.11). Considérons alors, comme dans [4] et [41] pour les polynômes orthogonaux, la matrice $\boldsymbol{R}_k$, matrice résolvante de $\boldsymbol{J}_k$ (dont la définition a été rappelée dans la Première Partie, Section 1).

$$\boldsymbol{R}_k(x) = (x\boldsymbol{I}_k - \boldsymbol{J}_k)^{-1}$$

Alors, nous allons tout d'abord énoncer un premier résultat concernant ces matrices résolvantes.

Lemme 4.1

Soit $\boldsymbol{Q}_k$ la matrice définie à la Proposition 2.6. Soit $\boldsymbol{Z}_k$ la matrice diagonale composée des zéros de P_k et $\boldsymbol{R}_k$ la matrice résolvante de $\boldsymbol{J}_k$.

Alors, si les zéros de P_k sont deux à deux distincts, on a

$$\boldsymbol{R}_k(x) = \boldsymbol{Q}_k(x\boldsymbol{I}_k - \boldsymbol{Z}_k)^{-1}\boldsymbol{Q}_k^{-1}.$$

Preuve :

De la Proposition 2.6 on a $\boldsymbol{J}_k = \boldsymbol{Q}_k\boldsymbol{Z}_k\boldsymbol{Q}_k^{-1}$, où $\boldsymbol{Q}_k$ est inversible puisque les zéros de P_k sont distincts.
Alors

$$\begin{aligned}(x\boldsymbol{I}_k - \boldsymbol{J}_k)^{-1} &= (x\boldsymbol{I}_k - \boldsymbol{Q}_k\boldsymbol{Z}_k\boldsymbol{Q}_k^{-1})^{-1} \\ &= \boldsymbol{Q}_k(x\boldsymbol{I}_k - \boldsymbol{Z}_k)^{-1}\boldsymbol{Q}_k^{-1}.\end{aligned}$$

Ceci achève la démonstration de ce Lemme.

■

Ce Lemme nous servira à caractériser plus précisément ces matrices résolvantes $\boldsymbol{R}_k$. On obtient tout d'abord le

Théorème 4.1

Soit $\boldsymbol{J}_k$ la matrice de Hessenberg inférieure introduite en (II.11) et soit $\boldsymbol{R}_k$ la matrice résolvante de $\boldsymbol{J}_k$.

Alors, si les zéros de P_k sont deux à deux distincts, on peut trouver des matrices $\boldsymbol{R}_i^{(k)}$ telles que

$$\boldsymbol{R}_k(x) = \sum_{i=1}^{k} \frac{1}{x - z_i^{(k)}} \boldsymbol{R}_i^{(k)} \tag{II.27}$$

avec

$$\boldsymbol{I}_k = \sum_{i=1}^{k} \boldsymbol{R}_i^{(k)} \tag{II.28}$$

et

$$\boldsymbol{J}_k = \sum_{i=1}^{k} \boldsymbol{R}_i^{(k)} z_i^{(k)} \tag{II.29}$$

Preuve :

$(x\boldsymbol{I}_k - \boldsymbol{Z}_k)$ est une matrice diagonale et son inverse est

$$\begin{pmatrix} (x - z_1^{(k)})^{-1} & & 0 \\ & \ddots & \\ 0 & & (x - z_k^{(k)})^{-1} \end{pmatrix} = \sum_{i=1}^{k} (x - z_i^{(k)})^{-1} \boldsymbol{E}_i^{(k)}$$

où la matrice $\boldsymbol{E}_i^{(k)}$ de dimension k contient un 1 sur la diagonale à la ligne i et des zéros partout ailleurs.

Ainsi, si les zéros de P_k sont deux à deux distincts, on peut appliquer le Lemme 4.1 et il s'en suit

$$\boldsymbol{R}_k(x) = \boldsymbol{Q}_k \sum_{i=1}^{k} (x - z_i^{(k)})^{-1} \boldsymbol{E}_i^{(k)} \boldsymbol{Q}_k^{-1} = \sum_{i=1}^{k} \frac{1}{x - z_i^{(k)}} \boldsymbol{Q}_k \boldsymbol{E}_i^{(k)} \boldsymbol{Q}_k^{-1}.$$

Ainsi, en posant

$$\boldsymbol{R}_i^{(k)} = \boldsymbol{Q}_k \boldsymbol{E}_i^{(k)} \boldsymbol{Q}_k^{-1}, \tag{II.30}$$

nous obtenons l'écriture (II.27).

Nous allons maintenant montrer que ces matrices vérifient les égalités (II.28) et (II.29).

La première égalité est évidente dans la mesure où $\sum_{i=1}^{k} \boldsymbol{E}_i^{(k)} = \boldsymbol{I}_k$, par définition des matrices $\boldsymbol{E}_i^{(k)}$.

La seconde se trouve aisément car $\boldsymbol{Z}_k = \sum_{i=1}^{k} z_i^{(k)} \boldsymbol{E}_i^{(k)}$.

L'expression des matrices $\boldsymbol{R}_i^{(k)}$ est alors déterminée par l'expression (II.30).

■

Enfin, on montre un dernier résultat concernant ces matrices résolvantes et leurs propriétés à travers un Lemme évident.

Lemme 4.2

Les matrices élémentaires $\boldsymbol{E}_i^{(k)}$ définies au Théorème 4.1 forment une famille de projecteurs orthogonaux.

Preuve :

Ce résultat est évident et n'est dû qu'à la forme particulière des matrices $\boldsymbol{E}_i^{(k)}$.

■

Théorème 4.2

Les matrices $\boldsymbol{R}_i^{(k)}$ définies au Théorème 4.1 forment une famille de projecteurs orthogonaux, c'est-à-dire

$$\boldsymbol{R}_i^{(k)2} = \boldsymbol{R}_i^{(k)}$$
$$\boldsymbol{R}_i^{(k)} \boldsymbol{R}_j^{(k)} = 0 \text{ pour tout } i \neq j.$$

Preuve :

Cela est simplement dû au fait que les matrices $\boldsymbol{E}_i^{(k)}$ forment elles aussi une famille de projecteurs orthogonaux.

Ainsi, on obtient

$$\boldsymbol{R}_i^{(k)} \boldsymbol{R}_j^{(k)} = \boldsymbol{Q}_k \boldsymbol{E}_i^{(k)} \boldsymbol{Q}_k^{-1} \boldsymbol{Q}_k \boldsymbol{E}_j^{(k)} \boldsymbol{Q}_k^{-1} = \boldsymbol{Q}_k \boldsymbol{E}_i^{(k)} \boldsymbol{E}_j^{(k)} \boldsymbol{Q}_k^{-1},$$

ce qui, en vertu du Lemme 4.2, achève la démonstration.

■

Il s'agit donc d'une généralisation, pour les polynômes biorthogonaux, de ce que l'on trouve dans [4] et qui a été donné dans la Première Partie, Section 1.

Un travail analogue sur les matrices $\widetilde{\boldsymbol{R}}_k(x)$ nous montre, bien évidemment, que les matrices $\widetilde{\boldsymbol{R}}_i^{(k)}$, définies de façon analogue aux matrices $\boldsymbol{R}_i^{(k)}$, sont elles-aussi des projecteurs orthogonaux.

On trouve également trivialement la

Proposition 4.1

Soit $\boldsymbol{J}_k$ la matrice de Hessenberg inférieure introduite en (II.11). Soient $\boldsymbol{R}_i^{(k)}$ les matrices définies au Théorème 4.1.

Alors l'inverse de la matrice résolvante vérifie

$$(x\boldsymbol{I}_k - \boldsymbol{J}_k) = \sum_{i=1}^{k} \left(x - z_i^{(k)}\right) \boldsymbol{R}_i^{(k)}.$$

Preuve :

Le résultat est évident si l'on forme $(x\boldsymbol{I}_k - \boldsymbol{J}_k)$ à l'aide des relations (II.28) et (II.29) et que l'on factorise chaque terme de la somme par $\boldsymbol{R}_i^{(k)}$.

■

Toutes ces relations sont en particulier valables pour les polynômes orthogonaux ainsi que pour les polynômes vectoriellement orthogonaux.

5 Méthode de bordage et biorthogonalité

Le but de cette Section est de montrer l'équivalence qui existe entre le calcul des polynômes biorthogonaux [8] (en utilisant les relations connues de ces derniers) et la méthode de bordage [34]. L'application de ce résultat sur des matrices particulières sera ensuite étudiée. Des résultats partiels de cette Section peuvent être trouvés dans [53].

Nous allons donc tout d'abord définir le cadre général dans lequel nous nous situerons dans la **première sous-section** ainsi qu'un bref rappel de la méthode de bordage pour les systèmes linéaires. Les polynômes biorthogonaux des Sections précédentes seront alors considérés et l'équivalence sera établie entre le calcul de ces derniers et la méthode de bordage.

Dans la **seconde sous-section**, nous nous consacrerons plus particulièrement à deux types de polynômes biorthogonaux particuliers que sont les polynômes orthogonaux et les polynômes orthogonaux de dimension -1 (voir par exemple [7] puis [23]). Respectivement, le cas des matrices de Hankel puis des matrices de Toeplitz sera alors abordé.

5.1 Cadre général

Définissons ici le contexte d'étude ainsi que les diverses relations qui nous seront utiles dans la suite.

Voyons ensuite les relations qui existent entre le calcul des polynômes biorthogonaux et la mise en œuvre de la méthode de bordage.

5.1.1 Contexte

Considérons le système linéaire d'équations

$$\begin{pmatrix} c_{0,1} & \cdots & c_{0,k} \\ \vdots & & \vdots \\ c_{k-1,1} & \cdots & c_{k-1,k} \end{pmatrix} \begin{pmatrix} a_1 \\ \vdots \\ a_k \end{pmatrix} = \begin{pmatrix} b_0 \\ \vdots \\ b_{k-1} \end{pmatrix} \tag{II.31}$$

où tous les coefficients considérés sont des complexes.

En posant $a_0 = 1$ et $c_{i,0} = -b_i$, le système devient

$$\begin{aligned} & a_0 = 1 \\ & a_0 c_{0,0} + a_1 c_{0,1} + \cdots + a_k c_{0,k} = 0 \\ & \cdots\cdots\cdots\cdots\cdots\cdots\cdots \\ & a_0 c_{k-1,0} + a_1 c_{k-1,1} + \cdots + a_k c_{k-1,k} = 0. \end{aligned} \tag{II.32}$$

Considérons maintenant, comme dans le cadre des polynômes biorthogonaux, les fonctionnelles $\mathcal{L}_i$ définies sur $\mathcal{P}$ par

$$\mathcal{L}_i(x^j) = c_{i,j} \text{ pour } i, j = 0, 1, \ldots$$

et définissons le polynôme de $\mathbb{C}[X]$

$$P_k(x) = a_0 + a_1 x + \cdots + a_k x^k.$$

Alors, en utilisant la définition du polynôme P_k et les égalités (II.32), le système d'équations précédent peut s'écrire

$$\begin{aligned} &P_k(0) = 1 \\ &\mathcal{L}_i(P_k) = 0 \qquad i = 0, \ldots, k-1. \end{aligned} \tag{II.33}$$

Pour s'en assurer, il suffit d'utiliser la linéarité des fonctionnelles $\mathcal{L}_i$. Cette nouvelle écriture est alors évidente.

Ainsi, résoudre (II.31) est équivalent au calcul des polynômes P_k, lorsque ceux-ci existent, qui satisfont (II.33).

D'après la définition introduite dans [8] et rappelée dans la Définition 1.1, il s'agit donc de calculer la famille des polynômes biorthogonaux à gauche P_k par rapport aux fonctionnelles $\mathcal{L}_i$.

On peut montrer (Brezinski [8]) que, si P_{k+1} est de degré $k+1$ exactement, alors il vérifie une relation du type

$$P_{k+1}(x) \;=\; P_k(x) - \lambda_k x P_k^{(1)}(x) \tag{II.34}$$

où $P_k^{(1)}$ est le polynôme unitaire de degré k exactement qui vérifie

$$\mathcal{L}_i(x P_k^{(1)}) = 0 \text{ pour } i = 0, \ldots, k-1 \tag{II.35}$$

et où

$$\lambda_k = \frac{\mathcal{L}_k(P_k)}{\mathcal{L}_k(x P_k^{(1)})}$$

avec $P_0(x) = P_0^{(1)}(x) = 1$.

Il suffit en effet d'appliquer $\mathcal{L}_i$ aux polynômes P_k définis à la relation (II.34) et de considérer les conditions d'orthogonalité que vérifient les polynômes $P_k^{(1)}$ en (II.35). L'unicité des polynômes biorthogonaux, à multiplication par une constante près, conduit alors à cette relation.

Ainsi, les solutions de (II.31) pour des valeurs de plus en plus grandes de k peuvent être calculées récursivement si les polynômes $P_k^{(1)}$ peuvent également être obtenus récursivement.

Comme nous le verrons plus loin, cela est possible dans certains cas. Mais, pour le moment, cherchons le rapport qui existe entre ce processus et la méthode de bordage.

5.1.2 La méthode de bordage

Rappelons tout d'abord brièvement la méthode de bordage ainsi que les notations que nous allons adopter dans la suite. Voyons dans un deuxième temps le rapport qu'il peut y avoir entre la méthode de bordage et les polynômes biorthogonaux tels qu'introduits dans la sous-section 5.1.

Considérons les deux systèmes linéaires d'équations d'ordre respectif k et $k+1$

$$\begin{aligned} \boldsymbol{A}_k z_k &= d_k \\ \boldsymbol{A}_{k+1} z_{k+1} &= d_{k+1} \end{aligned} \tag{II.36}$$

où

$$\boldsymbol{A}_{k+1} = \begin{pmatrix} \boldsymbol{A}_k & u_k \\ v_k & \delta_k \end{pmatrix} \text{ et } d_{k+1} = \begin{pmatrix} d_k \\ f_k \end{pmatrix},$$

$\boldsymbol{A}_k$ (resp. $\boldsymbol{A}_{k+1}$) étant une matrice carrée inversible de dimension k (resp. $k+1$), u_k et d_k deux vecteurs colonne à k composantes, v_k un vecteur ligne à k composantes, a_k et f_k deux scalaires pour que les expressions précédentes aient un sens.

La méthode de bordage [34] consiste à calculer récursivement le vecteur z_{k+1} introduit en (II.36) à partir de z_k par la relation

$$z_{k+1} = \begin{pmatrix} z_k \\ 0 \end{pmatrix} + \frac{f_k - v_k z_k}{\xi_k} \begin{pmatrix} -\boldsymbol{A}_k^{-1} u_k \\ 1 \end{pmatrix} \tag{II.37}$$

avec $\xi_k = \delta_k - v_k \boldsymbol{A}_k^{-1} u_k$.

Ainsi, on obtient le

Théorème 5.1

Si $\boldsymbol{A}_k$ désigne la matrice du système (II.32), d_k son second membre et z_k la solution associée et si l'on multiplie chaque côté de (II.37) scalairement par le vecteur $(1, x, \ldots, x^{k+1})^{\mathrm{T}}$ alors on obtient la relation polynomiale

$$P_{k+1}(x) = P_k(x) + \frac{f_k - v_k z_k}{\xi_k} x P_k^{(1)}(x)$$

où $P_k^{(1)}$, qui est déterminé à multiplication par une constante près, est choisi unitaire et l'on a de plus

$$\beta_k = \mathcal{L}_k(x P_k^{(1)})$$

et

$$v_k z_k = \mathcal{L}_k(P_k).$$

Preuve :

Montrons que le vecteur $\begin{pmatrix} -\boldsymbol{A}_k^{-1}u_k \\ 1 \end{pmatrix}$ admet zéro pour première composante. La première ligne de la matrice $\boldsymbol{A}_k$ considérée contient des zéros partout sauf dans la deuxième colonne. Son inverse aura alors la même caractéristique.

Compte tenu de la première ligne de $\boldsymbol{A}_k$ et comme le vecteur u_k admet zéro pour première composante, le produit $\boldsymbol{A}_k^{-1}u_k$ aura également un zéro comme première composante. Ainsi, la présence de $xP_k^{(1)}$ est justifiée. (Cela représente le produit scalaire du vecteur précédent par $(1, x, \ldots, x^{k+1})^{\mathrm{T}}$).

D'autre part, $-\boldsymbol{A}_k^{-1}u_k$ représente les coefficients de $1, x, \ldots, x^{k-1}$ de $P_k^{(1)}$ et par définition $\xi_k = c_{k,k+1} - v_k\boldsymbol{A}_k^{-1}u_k$. Comme $v_k = (c_{k,1}, \ldots, c_{k,k})$ et $\mathcal{L}_k(x^{k+1}) = c_{k,k+1}$, alors on obtient bien $\xi_k = \mathcal{L}_k(xP_k^{(1)})$.

D'un autre côté, $f_k = 0$ et $v_k z_k = \mathcal{L}_k(P_k)$ puisque z_k est le vecteur formé des coefficients de P_k.

■

D'où le

Corollaire 5.1.1

La mise en œuvre de la méthode de bordage est équivalente au calcul des polynômes biorthogonaux.

Ainsi, tout système linéaire d'équations peut être résolu récursivement par la méthode de bordage (voir [5] pour une procédure récursive de calcul des vecteurs $-\boldsymbol{A}_k^{-1}u_k$ et même [12] pour un sous-programme en Fortran qui la met en œuvre) ou par les relations de récurrence (II.34) portant sur une certaine famille de polynômes biorthogonaux.

La relation entre calcul de polynômes biorthogonaux et mise en œuvre de la méthode de bordage est donc immédiate.

Toutefois, pour que cette méthode soit d'un quelconque intérêt pratique, il est nécessaire de pouvoir calculer récursivement les polynômes $P_k^{(1)}$. Ceci n'est possible que s'il existe des relations simples entre les fonctionnelles $\mathcal{L}_i$, c'est-à-dire, en d'autres termes, s'il existe des relations entre les coefficients $c_{i,j}$ de la matrice du système considéré.

C'est pourquoi nous allons étudier deux cas particuliers : les matrices de Hankel et de Toeplitz, qui correspondent respectivement aux polynômes orthogonaux formels sur l'axe réel et sur le cercle unité. Des relations plus complexes entre les $c_{i,j}$ peuvent correspondre à une orthogonalité formelle sur des courbes algébriques [10] (voir par exemple les travaux de Marcellàn et al. [51] sur les lemniscates).

5.2 Cas particulier : les matrices de Hankel et Toeplitz

Dans cette sous-section, nous allons appliquer les résultats obtenus dans la sous-section 5.1 dans le cas où les fonctionnelles considérées $\mathcal{L}_i$ ont des propriétés particulières et sont liées par des relations simples.

Nous allons donc dans un premier temps étudier ce que deviennent les relations précédentes dans le cas des matrices de Hankel. Puis les matrices de Toeplitz seront considérées et les relations adaptées.

Cela nous mènera alors à la construction de divers algorithmes de résolution de systèmes associés à de telles matrices, qui seront équivalents au calcul des itérés de la méthode de bordage.

Enfin, la mise en œuvre de ces algorithmes sera étudiée sur quelques exemples et les résultats obtenus seront commentés.

5.2.1 Les matrices de Hankel

Considérons un cas particulier des polynômes biorthogonaux pour lequel des relations simples existent entre les fonctionnelles. Il s'agit des matrices de Hankel, liées naturellement aux polynômes orthogonaux.

Supposons que, $\forall i \geq 0$ et $\forall j \geq 2$

$$c_{i,j} = c_{i+1,j-1}$$

c'est-à-dire, en termes de relations fonctionnelles,

$$\mathcal{L}_i(x^j) = \mathcal{L}_{i+1}(x^{j-1}).$$

Puisque $c_{i,j}$ est un coefficient qui ne dépend, par définition, que de la somme $i+j$, on peut poser

$$c_{i,j} = c_{i+j}$$

et ainsi définir la fonctionnelle linéaire c sur $\mathcal{P}$ par

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

avec c_0 arbitraire.

Les matrices des systèmes successifs (II.31) sont alors des matrices de Hankel et ces systèmes successifs deviennent

$$\begin{pmatrix} c_1 & \cdots & c_k \\ \vdots & & \vdots \\ c_k & \cdots & c_{2k-1} \end{pmatrix} \begin{pmatrix} a_1 \\ \vdots \\ a_k \end{pmatrix} = - \begin{pmatrix} c_{0,0} \\ \vdots \\ c_{k-1,0} \end{pmatrix}. \tag{II.38}$$

Les matrices de Hankel considérées sont supposées inversibles $\forall k$ puisque c'est une condition nécessaire à la mise en œuvre de la méthode de bordage.

Remarque 5.1

Il ne faut toutefois pas oublier un point important. Nous avons en effet, par définition, $\mathcal{L}_i(1) = c_{i,0} = -b_i$ *et ainsi, dans ce cas particulier, on n'a pas l'égalité* $c_{i,0} = c_i = c_{m,n}$ *pour tout* m *et* n *tels que* $m+n = i$. *Alors, il vient que les* k *dernières équations du système (II.32) ne forment pas une matrice de Hankel rectangulaire.*

On voit facilement que les conditions de biorthogonalité des polynômes $P_k^{(1)}$ peuvent s'écrire

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

et ainsi $\{P_k^{(1)}\}_{k\geqslant 0}$ est la famille de polynômes orthogonaux formels (une généralisation des polynômes orthogonaux sur l'axe réel [4]) par rapport à la fonctionnelle $c^{(1)}$ définie par ses moments par

$$c^{(1)}(x^i) = c(x^{i+1}) = c_{i+1}.$$

Puisque les polynômes unitaires $P_k^{(1)}$ sont orthogonaux par rapport à la fonctionnelle $c^{(1)}$, ils satisfont la relation de récurrence à trois termes donnée au Théorème 1.2

$$P_{k+1}^{(1)}(x) = (x+\alpha_k)P_k^{(1)}(x) - \beta_k P_{k-1}^{(1)}(x) \tag{II.39}$$

avec $P_{-1}^{(1)}(x) = 0, P_0^{(1)}(x) = 1$ et

$$\begin{aligned} \beta_k &= \frac{c^{(1)}(x^k P_k^{(1)})}{c^{(1)}(x^{k-1}P_{k-1}^{(1)})} \\ \alpha_k &= \frac{\beta_k c^{(1)}(x^k P_{k-1}^{(1)}) - c^{(1)}(x^{k+1}P_k^{(1)})}{c^{(1)}(x^k P_k^{(1)})} \end{aligned} \tag{II.40}$$

si l'on utilise les notations adoptées dans cette Section.

Transformons maintenant les relations de récurrence précédentes en une procédure de résolution du système (II.31) quand la matrice est une matrice de Hankel.

On pose

$$\begin{aligned} P_k(x) &= a_0^{(k)} + \cdots + a_k^{(k)}x^k \text{ avec } a_0^{(k)} = 1, \\ P_k^{(1)}(x) &= b_0^{(k)} + \cdots + b_k^{(k)}x^k \text{ avec } b_k^{(k)} = 1. \end{aligned}$$

Ainsi (II.34) nous donne immédiatement

$$\begin{aligned} a_0^{(k+1)} &= 1 \\ a_i^{(k+1)} &= a_i^{(k)} - \lambda_k b_{i-1}^{(k)} \quad \text{pour } i = 1, \ldots, k \\ a_{k+1}^{(k+1)} &= -\lambda_k b_k^{(k)} \end{aligned} \tag{II.41}$$

avec

$$\lambda_k = \frac{\mathcal{L}_k(P_k)}{\mathcal{L}_k(xP_k^{(1)})} = \frac{-a_0^{(k)}b_k + a_1^{(k)}c_{k+1} + \cdots + a_k^{(k)}c_{2k}}{b_0^{(k)}c_{k+1} + \cdots + b_k^{(k)}c_{2k+1}}.$$

De même, (II.39) donne

$$\begin{aligned}
b_0^{(k+1)} &= \alpha_k b_0^{(k)} - \beta_k b_0^{(k-1)} \\
b_i^{(k+1)} &= b_{i-1}^{(k)} + \alpha_k b_i^{(k)} - \beta_k b_i^{(k-1)} \text{ pour } i = 1, \ldots, k-1 \\
b_k^{(k+1)} &= b_{k-1}^{(k)} + \alpha_k b_k^{(k)} \\
b_{k+1}^{(k+1)} &= 1.
\end{aligned}$$

Les coefficients α_k et β_k sont calculés par les relations (II.40) avec, par définition, $c^{(1)}(x^i P_k^{(1)}) = b_0^{(k)}c_{i+1} + \cdots + b_k^{(k)}c_{k+i+1}$.

Donnons maintenant une autre relation pour calculer le polynôme $P_{k+1}^{(1)}$. On considère à nouveau le système (II.38), mais avec un second membre où $c_{i,0}$ est remplacé par $c_i, \forall i$. C'est-à-dire que l'on obtient le système

$$\begin{pmatrix} c_1 & \cdots & c_k \\ \vdots & & \vdots \\ c_k & \cdots & c_{2k-1} \end{pmatrix} \begin{pmatrix} t_1 \\ \vdots \\ t_k \end{pmatrix} = - \begin{pmatrix} c_0 \\ \vdots \\ c_{k-1} \end{pmatrix}.$$

Dans ce système, c_0 peut être choisi arbitrairement. Soit alors Q_k le polynôme défini par

$$Q_k(x) = 1 + t_1 x + \cdots + t_k x^k$$

où les coefficients t_i dépendent de k. D'après la méthode de bordage, on peut prouver, comme plus haut, que si Q_{k+1} est de degré $k+1$ exactement, il existe un polynôme unitaire $Q_k^{(1)}$ de degré k exactement tel que

$$Q_{k+1}(x) = Q_k(x) - \lambda'_k x Q_k^{(1)}(x) \tag{II.42}$$

avec, comme précédemment,

$$\lambda'_k = \frac{\mathcal{L}'_k(Q_k)}{\mathcal{L}'_k(xQ_k^{(1)})}$$

et où les fonctionnelles $\mathcal{L}'_i$ sont définies, $\forall i, j \geq 0$, par

$$\mathcal{L}'_i(x^j) = c_{i+j}.$$

On a $Q_k(0) = 1$ et l'on vérifie aisément que les conditions d'orthogonalité

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

sont satisfaites.

Ainsi, Q_k et $Q_k^{(1)}$ sont les polynômes orthogonaux formels définis dans [4] et l'on peut prouver [11] que les polynômes $Q_k^{(1)}$ se calculent récursivement par la relation

$$Q_{k+1}^{(1)}(x) = \theta_k Q_{k+1}(x) + \gamma_k Q_k^{(1)}(x) \tag{II.43}$$

où θ_k et γ_k sont solutions de

$$\begin{aligned} &\theta_k t_{k+1} = 1 \\ &\theta_k c(x^{k+1} Q_{k+1}) + \gamma_k c^{(1)}(x^k Q_k^{(1)}) = 0 \end{aligned}$$

t_{k+1} étant le coefficient de x^{k+1} dans Q_{k+1}.

Proposition 5.1

Les polynômes $P_k^{(1)}$ et $Q_k^{(1)}$ sont identiques.

Preuve :

Les polynômes $P_k^{(1)}$ et $Q_k^{(1)}$ vérifient

$$c^{(1)}(x^i P_k^p(1)) = c^{(1)}(x^i Q_k^{(1)}) = 0 \text{ pour } 0 \leqslant i \leqslant k-1.$$

Les matrices de Hankel $\boldsymbol{A}_k$ étant inversibles $\forall k$, il y a existence et unicité des polynômes orthogonaux formels unitaires (puisqu'alors $\deg(P_k^{(1)}) = \deg(Q_k^{(1)}) = k$) relativement à la fonctionnelle c.

De plus, de la même façon que l'on a montré que les polynômes $P_k^{(1)}$ étaient unitaires, on montre que les polynômes $Q_k^{(1)}$ le sont également. Par unicité, ils sont donc identiques.

■

Il s'en suit, d'après (II.43), que l'on a

$$P_{k+1}^{(1)}(x) = \theta_k Q_{k+1}(x) + \gamma_k P_k^{(1)}(x). \tag{II.44}$$

Puisque les polynômes Q_k sont orthogonaux par rapport à la fonctionnelle c, ils peuvent être calculés directement par leur relation à trois termes, sans utiliser les polynômes $P_k^{(1)}$.

Les polynômes Q_k étant orthogonaux mais non unitaires, ils satisfont une relation de récurrence à trois termes (Brezinski [4]) de la forme

$$Q_{k+1}(x) = (\mu_k x + \nu_k) Q_k(x) - \omega_k Q_{k-1}(x) \tag{II.45}$$

avec $Q_{-1}(x) = 0, Q_0(x) = 1$ et

$$\begin{aligned} &\nu_k - \omega_k = 1 \\ &\mu_k c(x^k Q_k) - \omega_k c(x^{k-1} Q_{k-1}) = 0 \\ &\mu_k c(x^{k+1} Q_k) + \nu_k c(x^k Q_k) - \omega_k c(x^k Q_{k-1}) = 0. \end{aligned} \tag{II.46}$$

Effectuons maintenant un changement de notations et posons

$$Q_k(x) = q_0^{(k)} + \cdots + q_k^{(k)} x^k$$

avec $q_0^{(k)} = 1$.

De (II.44), on obtient

$$\begin{aligned} &b_i^{(k+1)} = \theta_k q_i^{(k+1)} + \gamma_k b_i^{(k)} \qquad \text{pour } i = 0, \ldots, k \\ &b_{k+1}^{(k+1)} = 1 \end{aligned}$$

avec $\theta_k = 1/q_{k+1}^{(k+1)}$.

D'après (II.45), on a

$$\begin{aligned} q_0^{(k+1)} &= 1 \\ q_i^{(k+1)} &= \mu_k q_{i-1}^{(k)} + \nu_k q_i^{(k)} - \omega_k q_i^{(k-1)} \text{ pour } i = 1, \ldots, k-1 \\ q_k^{(k+1)} &= \mu_k q_{k-1}^{(k)} + \nu_k q_k^{(k)} \\ q_{k+1}^{(k+1)} &= \mu_k q_k^{(k)}. \end{aligned}$$

Les coefficients μ_k, ν_k et ω_k s'obtiennent à partir de (II.46) en remarquant que $c(x^i Q_k) = q_0^{(k)} c_i + \cdots + q_k^{(k)} c_{i+k}$.

Finalement (II.42) nous donne

$$\begin{aligned} &q_0^{(k+1)} = 1 \\ &q_i^{(k+1)} = q_i^{(k)} - \lambda'_k b_{i-1}^{(k)} \quad \text{pour } i = 1, \ldots, k \\ &q_{k+1}^{(k+1)} = -\lambda'_k b_k^{(k)} \end{aligned}$$

avec

$$\lambda'_k = \frac{\mathcal{L}'_k(Q_k)}{\mathcal{L}'_k(xP_k^{(1)})} = \frac{q_0^{(k)} c_k + \cdots + q_k^{(k)} c_{2k}}{b_0^{(k)} c_{k+1} + \cdots + b_k^{(k)} c_{2k+1}}.$$

Ainsi, en combinant les diverses possibilités de calcul des polynômes P_k, Q_k et $P_k^{(1)}$, trois algorithmes peuvent être obtenus pour résoudre un système de Hankel.

Tout d'abord, on peut utiliser les relations (II.34) et (II.39). On obtient ainsi un premier algorithme, noté H1.

Algorithme H1($\boldsymbol{A}, \boldsymbol{b}$)

- **Initialisations**

$a_0^{(0)} = b_0^{(0)} = 1$

- **Itérations**

Pour $k = 0, \ldots, n-1$ **Faire**

$\lambda_k \leftarrow \left(-a_0^{(k)} b_k + \sum_{i=1}^{k} a_i^{(k)} c_{k+i}\right) / \left(\sum_{i=0}^{k} b_i^{(k)} c_{k+i+1}\right)$

$a_0^{(k+1)} = 1$

Pour $i = 1, \ldots, k$ **Faire**

$a_i^{(k+1)} \leftarrow a_i^{(k)} - \lambda_k b_{i-1}^{(k)}$

Fin de Pour.

$a_{k+1}^{(k+1)} \leftarrow -\lambda_k b_k^{(k)}$

$\beta_k \leftarrow \left(\sum_{i=0}^{k} c_{k+i} b_i^{(k)}\right) / \left(\sum_{i=0}^{k-1} c_{k+i-1} b_i^{(k-1)}\right)$

$\alpha_k \leftarrow \left(\beta_k \sum_{i=0}^{k-1} c_{k+i} b_i^{(k-1)} - \sum_{i=0}^{k} c_{k+i+2} b_i^{(k)}\right) / \left(\sum_{i=0}^{k} c_{k+i+1} b_i^{(k)}\right)$

$b_0^{(k+1)} \leftarrow \alpha_k b_0^{(k)} - \beta_k b_0^{(k-1)}$

Pour $i = 1, \ldots, k-1$ **Faire**

$b_i^{(k+1)} \leftarrow b_{i-1}^{(k)} + \alpha_k b_i^{(k)} - \beta_k b_i^{(k-1)}$

Fin de Pour.

$b_k^{(k+1)} \leftarrow b_{k-1}^{(k)} + \alpha_k$

$b_{k+1}^{(k+1)} = 1$

Fin de Pour.

Puis, en utilisant les relations (II.34), (II.42) et (II.44), on obtient un deuxième algorithme, que l'on note H2.

Algorithme H2($\boldsymbol{A}, \boldsymbol{b}, c_0$)

- **Initialisations**

$a_0^{(0)} = b_0^{(0)} = q_0^{(0)} = 1$

- **Itérations**

Pour $k = 0, \ldots, n-1$ **Faire**

$\lambda_k \leftarrow \left(-a_0^{(k)} b_k + \sum_{i=1}^{k} a_i^{(k)} c_{k+i}\right) / \left(\sum_{i=0}^{k} b_i^{(k)} c_{k+i+1}\right)$

$\lambda'_k \leftarrow \left(\sum_{i=0}^{k} q_i^{(k)} c_{k+i}\right) / \left(\sum_{i=0}^{k} b_i^{(k)} c_{k+i+1}\right)$

$a_0^{(k+1)} = q_0^{(k+1)} = 1$

Pour $i = 1, \ldots, k$ **Faire**

$a_i^{(k+1)} \leftarrow a_i^{(k)} - \lambda_k b_{i-1}^{(k)}$
$q_i^{(k+1)} \leftarrow q_i^{(k)} - \lambda'_k b_{i-1}^{(k)}$
Fin de Pour.
$a_{k+1}^{(k+1)} \leftarrow -\lambda_k b_k^{(k)}$
$q_{k+1}^{(k+1)} \leftarrow -\lambda'_k b_k^{(k)}$
$\theta_k \leftarrow 1/q_{k+1}^{(k+1)}$
$\gamma_k \leftarrow -\theta_k \left(\sum_{i=0}^{k+1} c_{k+i+1} q_i^{(k+1)}\right) / \left(\sum_{i=0}^{k} c_{k+i+1} b_i^{(k)}\right)$
Pour $i = 0, \ldots, k$ **Faire**
$b_i^{(k+1)} \leftarrow \theta_k q_i^{(k+1)} + \gamma_k b_i^{(k)}$
Fin de Pour.
$b_{k+1}^{(k+1)} = 1$
Fin de Pour.

Et enfin, si l'on utilise les relations (II.34), (II.44) et (II.45), le dernier algorithme, noté H3, s'en suit.

Algorithme H3($\boldsymbol{A}, \boldsymbol{b}, c_0$)

- **Initialisations**
  $a_0^{(0)} = b_0^{(0)} = q_0^{(0)} = 1$
- **Itérations**
  Pour $k = 0, \ldots, n-1$ **Faire**
  $\lambda_k \leftarrow (-a_0^{(k)} b_k + \sum_{i=1}^{k} a_i^{(k)} c_{k+i}) / (\sum_{i=0}^{k} b_i^{(k)} c_{k+i+1})$
  $a_0^{(k+1)} = 1$
  Pour $i = 1, \ldots, k$ **Faire**
  $a_i^{(k+1)} = a_i^{(k)} - \lambda_k b_{i-1}^{(k)}$
  Fin de Pour.
  $a_{k+1}^{(k+1)} \leftarrow -\lambda_k b_k^{(k)}$
  $q_0^{(k+1)} = 1$
  Calcul[1] de μ_k, ν_k et ω_k
  Pour $i = 1, \ldots, k-1$ **Faire**
  $q_i^{(k+1)} \leftarrow \mu_k q_{i-1}^{(k)} + \nu_k q_i^{(k)} - \omega_k q_i^{(k-1)}$
  Fin de Pour.
  $q_k^{(k+1)} \leftarrow \mu_k q_{k-1}^{(k)} + \nu_k q_k^{(k)}$
  $q_{k+1}^{(k+1)} \leftarrow \mu_k q_k^{(k)}$
  $\theta_k \leftarrow 1/q_{k+1}^{(k+1)}$
  $\gamma_k \leftarrow -\theta_k \left(\sum_{i=0}^{k+1} c_{k+i+1} q_i^{(k+1)}\right) / \left(\sum_{i=0}^{k} c_{k+i+1} b_i^{(k)}\right)$
  Pour $i = 0, \ldots, k$ **Faire**
  $b_i^{(k+1)} \leftarrow \theta_k q_i^{(k+1)} + \gamma_k b_i^{(k)}$

[1] Le calcul des coefficients μ_k, ν_k et ω_k n'est pas explicité ici pour des raisons évidentes de clarté. Il se déduit sans peine de la résolution du système linéaire 3×3 sous-jacent.

Fin de Pour.
$b_{k+1}^{(k+1)} = 1$
Fin de Pour.

Le tableau suivant mentionne les besoins de chaque algorithme du point de vue calculatoire (Flops) en reprenant le nombre d'opérations nécessaire à chaque itération (itération k). Il consigne également l'encombrement mémoire requis (mémoire) en donnant le nombre de vecteurs nécessaires à chaque algorithme (il s'agit du nombre total puisqu'il n'est pas utile de stocker tous les vecteurs).

Algorithme	H1	H2	H3[2]
Flops	$20k + 11$	$19k + 22$	$18k + 16$
Mémoire	3	4	4

TAB. 1: *Emplacement mémoire général et coût opératoire requis par itération pour la mise en œuvre des algorithmes H1, H2 et H3.*

Ainsi, chaque algorithme est un algorithme qui requiert $O(n^2)$ opérations. Par exemple, H1 demande $\sum_{i=1}^{n} 20k + 11 = 10n^2 + 21n$ opérations.

L'algorithme qui nécessite le moins d'opérations est H3 alors que celui qui en demande le plus est H1. H2 se situe entre les deux. Du point de vue encombrement mémoire, le premier algorithme (H1) requiert un vecteur de moins que les deux autres.

Il faut noter que, si certains des polynômes P_k, Q_k ou $P_k^{(1)}$ n'existent pas, alors une division par zéro se produit dans une des relations de récurrence. Une telle situation est appelée *true breakdown.*

Une division par zéro peut aussi subvenir même si tous les polynômes existent. Cette situation, connue sous le nom de *ghost breakdown*, se produit quand les relations de récurrence considérées ne peuvent pas être utilisées. Ce point est développé en détail dans [13] et de tels problèmes peuvent être évités en utilisant des relations de récurrence spécifiques [11] qui proviennent d'une méthode de bordage par blocs dans laquelle on rajoute simultanément plusieurs lignes et plusieurs colonnes à la matrice.

Il en va de même pour ce que l'on appelle *near-breakdown* qui est dû à une division par un nombre voisin de zéro et qui est la cause d'instabilité numérique.

5.2.2 Les matrices de Toeplitz

D'autres fonctionnelles bénéficiant de relations simples de récurrence vont être maintenant considérées. Il s'agit des fonctionnelles liées aux matrices de Toeplitz

[2]L'inversion de la matrice 3×3 n'est pas prise en compte

et dont les polynômes sont orthogonaux de dimension -1.

Considérons désormais le cas où

$$c_{i,j} = c_{i+1,j+1}$$

qui correspond à

$$\mathcal{L}_i(x^j) = \mathcal{L}_{i+1}(x^{j+1}).$$

Puisque $c_{i,j}$ ne dépend que de la différence $i-j$, on peut poser, de façon similaire au cas des matrices de Hankel,

$$c_{i,j} = c_{i-j}$$

et l'on définit la fonctionnelle linéaire c , cette fois-ci sur l'espace des polynômes de Laurent, en posant

$$c(x^i) = c_i, \ i \in \mathbb{Z}.$$

Les matrices des systèmes successifs (II.31) sont alors des matrices de Toeplitz. Comme dans le cas Hankel, $\mathcal{L}_i(1) = -b_i$ et l'on considère alors le système

$$\begin{pmatrix} c_{-1} & \cdots & c_{-k} \\ c_0 & \cdots & c_{-k+1} \\ \vdots & & \vdots \\ c_{k-2} & \cdots & c_{-1} \end{pmatrix} \begin{pmatrix} t_1 \\ \vdots \\ t_k \end{pmatrix} = - \begin{pmatrix} c_0 \\ \vdots \\ c_{k-1} \end{pmatrix}. \tag{II.47}$$

Dans ce système, c_{k-1} est choisi de façon arbitraire.

Soit Q_k le polynôme

$$Q_k(x) = 1 + t_1 x + \cdots + t_k x^k$$

où les coefficients t_i dépendent de k.

Par la méthode de bordage, on peut prouver, comme plus haut, que si Q_{k+1} est de degré $k+1$ exactement, il existe un polynôme unitaire $Q_k^{(1)}$ de degré k tel que

$$Q_{k+1}(x) = Q_k(x) - \lambda'_k x Q_k^{(1)}(x) \tag{II.48}$$

avec

$$\lambda'_k = \frac{\mathcal{L}'_k(Q_k)}{\mathcal{L}'_k(xQ_k^{(1)})}$$

et où les fonctionnelles $\mathcal{L}'_i$ sont définies, $\forall i, j \geq 0$, par

$$\mathcal{L}'_i(x^j) = c_{i-j}.$$

On peut prouver, comme plus haut, que $Q_k^{(1)}$ est identique à $P_k^{(1)}$ et que l'on a

$$c(x^{i-k}\widetilde{Q}_k) = c(x^{i-k-1}\widetilde{P}_k^{(1)}) = 0 \text{ pour } i = 0, \ldots, k-1$$

où les polynômes $\widetilde{Q}_k$ et $\widetilde{P}_k$ sont définis par $\widetilde{Q}_k(x) = x^k Q_k(x^{-1})$ et $\widetilde{P}_k^{(1)}(x) = x^k P_k^{(1)}(x^{-1})$ (voir [7]).

Les polynômes Q_k et $P_k^{(1)}$ sont des polynômes orthogonaux de dimension -1 (une généralisation des polynômes orthogonaux sur le cercle unité) et l'on peut prouver [7] que les polynômes $P_k^{(1)}$ s'obtiennent récursivement par la relation

$$P_{k+1}^{(1)}(x) = \rho_k Q_{k+1}(x) + \eta_k Q_k(x) \tag{II.49}$$

où ρ_k et η_k sont solutions de

$$\begin{aligned} &\rho_k t_{k+1} = 1 \\ &\rho_k c(x^{-k-2}\widetilde{Q}_{k+1}) + \eta_k c(x^{-k-1}\widetilde{Q}_k) = 0, \end{aligned}$$

t_{k+1} étant le coefficient de x^{k+1} dans Q_{k+1}.

Puisque les polynômes $P_k^{(1)}$ sont orthogonaux de dimension -1, ils satisfont une relation de récurrence à trois termes de la forme (noter le x devant $P_{k-1}^{(1)}$)

$$P_{k+1}^{(1)}(x) = (x + \alpha_k)P_k^{(1)}(x) - \beta_k x P_{k-1}^{(1)}(x) \tag{II.50}$$

avec $P_{-1}^{(1)}(x) = 0, P_0^{(1)}(x) = 1$ et

$$\begin{aligned} c(x^{-k-2}\widetilde{P}_k^{(1)}) &= \beta_k c(x^{-k-1}\widetilde{P}_{k-1}^{(1)}) \\ \alpha_k c(x^{-1}\widetilde{P}_k^{(1)}) &= \beta_k c(x^{-1}\widetilde{P}_{k-1}^{(1)}). \end{aligned}$$

Les polynômes Q_k sont orthogonaux de dimension -1 et, ainsi, ils peuvent être calculés directement sans utiliser les polynômes $P_k^{(1)}$.

On a

$$Q_{k+1}(x) = (\mu_k x + 1)Q_k(x) - \omega_k x Q_{k-1}(x) \tag{II.51}$$

avec $Q_{-1}(x) = 0$, $Q_0(x) = 1$ et

$$\begin{aligned} \mu_k c(x^{-k-1}\widetilde{Q}_k) &= \omega_k c(x^{-k}\widetilde{Q}_{k-1}) \\ c(\widetilde{Q}_k) &= \omega_k c(\widetilde{Q}_{k-1}). \end{aligned}$$

Utilisons maintenant ces relations de récurrence pour la résolution du système (II.31) quand la matrice est de Toeplitz.

(II.34) nous donne à nouveau les relations (II.41) où $\lambda_k = \dfrac{\mathcal{L}_k(P_k)}{\mathcal{L}_k(xP_k^{(1)})}$ avec

$$\begin{aligned} \mathcal{L}_k(P_k) &= -a_0^{(k)}b_k + a_1^{(k)}c_{k-1} + \ldots + a_k^{(k)}c_0 \\ \text{et } \mathcal{L}_k(xP_k^{(1)}) &= b_0^{(k)}c_{k-1} + \ldots + b_{k-1}^{(k)}c_0 + b_k^{(k)}c_{-1}. \end{aligned}$$

Nous posons désormais

$$Q_k(x) = q_0^{(k)} + q_1^{(k)}x + \cdots + q_k^{(k)}x^k.$$

Nous obtenons, d'après (II.48)

$$\begin{aligned} q_0^{(k+1)} &= 1 \\ q_i^{(k+1)} &= q_i^{(k)} - \lambda'_k b_{i-1}^{(k)} \text{ pour } i = 1, \ldots, k \\ q_{k+1}^{(k+1)} &= -\lambda'_k b_k^{(k)} \end{aligned}$$

avec

$$\lambda'_k = \frac{\mathcal{L}'_k(Q_k)}{\mathcal{L}'_k(xP_k^{(1)})} = \frac{q_0^{(k)}c_k + \cdots + q_k^{(k)}c_0}{b_0^{(k)}c_{k-1} + \cdots + b_k^{(k)}c_{-1}}.$$

(II.49) nous donne

$$\begin{aligned} b_i^{(k+1)} &= \rho_k q_i^{(k+1)} + \eta_k q_i^{(k)} \text{ pour } i = 0, \ldots, k \\ b_{k+1}^{(k+1)} &= 1 \end{aligned}$$

avec $\rho_k = 1/q_{k+1}^{(k+1)}$.

De la relation (II.50) on obtient

$$\begin{aligned} b_0^{(k+1)} &= \alpha_k b_0^{(k)} \\ b_i^{(k+1)} &= b_{i-1}^{(k)} + \alpha_k b_i^{(k)} - \beta_k b_{i-1}^{(k-1)} \text{ pour } i = 1, \ldots, k \\ b_{k+1}^{(k+1)} &= 1. \end{aligned}$$

Finalement, la relation (II.51) donne

$$\begin{aligned} q_0^{(k+1)} &= 1 \\ q_i^{(k+1)} &= \mu_k q_{i-1}^{(k)} + q_i^{(k)} - \omega_k q_{i-1}^{(k-1)} \text{ pour } i = 1, \ldots, k \\ q_{k+1}^{(k+1)} &= \mu_k q_k^{(k)} \end{aligned}$$

avec

$$\omega_k = \frac{\mathcal{L}'_k(Q_k)}{\mathcal{L}'_{k-1}(Q_{k-1})}$$

où $\mathcal{L}'_k(Q_k)$ est obtenu avec les relations précédentes.

On peut ainsi obtenir trois algorithmes pour la résolution des systèmes de Toeplitz qui seront équivalents à la mise en œuvre de la méthode de bordage pour de tels systèmes.

Ainsi, si l'on utilise les relations (II.34) et (II.51), on trouve un premier algorithme, que l'on note T1.

Algorithme T1($\boldsymbol{A}, \boldsymbol{b}$)

- **Initialisations**

  $a_0^{(0)} = b_0^{(0)} = 1$

- **Itérations**

  Pour $k = 0, \ldots, n-1$ **Faire**

  $\lambda_k \leftarrow \left(-a_0^{(k)} b_k + \sum_{i=1}^{k} a_i^{(k)} c_{k-i}\right) / \left(\sum_{i=0}^{k} b_i^{(k)} c_{k-i-1}\right)$

  $a_0^{(k+1)} = 1$

  Pour $i = 1, \ldots, k$ **Faire**

  $a_i^{(k+1)} \leftarrow a_i^{(k)} - \lambda_k b_{i-1}^{(k)}$

  Fin de Pour.

  $a_{k+1}^{(k+1)} \leftarrow -\lambda_k b_k^{(k)}$

  $\beta_k \leftarrow \left(\sum_{i=0}^{k} b_i^{(k)} c_{-i-2}\right) / \left(\sum_{i=0}^{k-1} b_i^{(k-1)} c_{-i-2}\right)$

  $\alpha_k \leftarrow \beta_k \left(\sum_{i=0}^{k-1} b_i^{(k-1)} c_{k-2-i}\right) / \left(\sum_{i=0}^{k} b_i^{(k)} c_{k-i-1}\right)$

  $b_0^{(k+1)} \leftarrow \alpha_k b_0^{(k)}$

  Pour $i = 1, \ldots, k$ **Faire**

  $b_i^{(k+1)} \leftarrow b_{i-1}^{(k)} + \alpha_k b_i^{(k)} - \beta_k b_{i-1}^{(k-1)}$

  Fin de Pour.

  $b_{k+1}^{(k+1)} = 1$

  Fin de Pour.

Un deuxième algorithme peut être obtenu lorsque l'on considère les relations (II.34), (II.48) et (II.49). Il est noté T2.

Algorithme T2($\boldsymbol{A}, \boldsymbol{b}$)

- **Initialisations**

  $a_0^{(0)} = b_0^{(0)} = q_0^{(0)} = 1$

- **Itérations**

  Pour $k = 0, \ldots, n-1$ **Faire**

  $\lambda_k \leftarrow \left(-a_0^{(k)} b_k + \sum_{i=1}^{k} a_i^{(k)} c_{k-i}\right) / \left(\sum_{i=0}^{k} b_i^{(k)} c_{k-i-1}\right)$

  $\lambda'_k \leftarrow \left(\sum_{i=0}^{k} q_i^{(k)} c_{k-i}\right) / \left(\sum_{i=0}^{k} b_i^{(k)} c_{k-i-1}\right)$

  $a_0^{(k+1)} = q_0^{(k+1)} = 1$

  Pour $i = 1, \ldots, k$ **Faire**

  $a_i^{(k+1)} \leftarrow a_i^{(k)} - \lambda_k b_{i-1}^{(k)}$

  $q_i^{(k+1)} \leftarrow q_i^{(k)} - \lambda'_k b_{i-1}^{(k)}$

  Fin de Pour.

$a_{k+1}^{(k+1)} \leftarrow -\lambda_k b_k^{(k)}$
$q_{k+1}^{(k+1)} \leftarrow -\lambda'_k b_k^{(k)}$
$\rho_k \leftarrow 1/q_{k+1}^{(k+1)}$
$\eta_k \leftarrow -\rho_k \left(\sum_{i=0}^{k+1} q_i^{(k+1)} c_{-i-1}\right) / \left(\sum_{i=0}^{k} q_i^{(k)} c_{-i-1}\right)$
Pour $i = 0, \ldots, k$ **Faire**
$b_i^{(k+1)} \leftarrow \rho_k q_i^{(k+1)} + \eta_k q_i^{(k)}$
Fin de Pour.
$b_{k+1}^{(k+1)} = 1$
Fin de Pour.

Enfin, un troisième et dernier algorithme peut être donné en utilisant les relations (II.34), (II.49) et (II.51). On le note T3.

Algorithme T3($\boldsymbol{A}, \boldsymbol{b}$)

- **Initialisations**
  $a_0^{(0)} = b_0^{(0)} = q_0^{(0)} = 1$
- **Itérations**
  Pour $k = 0, \ldots, n-1$ **Faire**
  $\lambda_k \leftarrow \left(-a_0^{(k)} b_k + \sum_{i=1}^{k} a_i^{(k)} c_{k-i}\right) / \left(\sum_{i=0}^{k} b_i^{(k)} c_{k-i-1}\right)$
  $a_0^{(k+1)} = 1$
  Pour $i = 1, \ldots, k$ **Faire**
  $a_i^{(k+1)} \leftarrow a_i^{(k)} - \lambda_k b_{i-1}^{(k)}$
  Fin de Pour.
  $a_{k+1}^{(k+1)} \leftarrow -\lambda_k b_k^{(k)}$
  $\omega_k \leftarrow \left(\sum_{i=0}^{k} q_i^{(k)} c_{k-i}\right) / \left(\sum_{i=0}^{k-1} q_i^{(k-1)} c_{k-i-1}\right)$
  $\mu_k \leftarrow \omega_k \left(\sum_{i=0}^{k-1} q_i^{(k-1)} c_{-i-1}\right) / \left(\sum_{i=0}^{k} q_i^{(k)} c_{-i-1}\right)$
  $q_0^{(k+1)} = 1$
  Pour $i = 1, \ldots, k$ **Faire**
  $q_i^{(k+1)} \leftarrow \mu_k q_{i-1}^{(k)} + q_i^{(k)} - \omega_k q_{i-1}^{(k-1)}$
  Fin de Pour.
  $q_{k+1}^{(k+1)} \leftarrow \mu_k q_k^{(k)}$
  $\rho_k \leftarrow 1/q_{k+1}^{(k+1)}$
  $\eta_k \leftarrow -\rho_k \left(\sum_{i=0}^{k+1} q_i^{(k+1)} c_{-i-1}\right) / \left(\sum_{i=0}^{k} q_i^{(k)} c_{-i-1}\right)$
  Pour $i = 0, \ldots, k$ **Faire**
  $b_i^{(k+1)} \leftarrow \rho_k q_i^{(k+1)} + \eta_k q_i^{(k)}$
  Fin de Pour.
  $b_{k+1}^{(k+1)} = 1$
  Fin de Pour.

Le tableau suivant mentionne les besoins de chaque algorithme du point de vue calculatoire (Flops) en reprenant le nombre d'opérations nécessaire à chaque itération (itération k). Il consigne également l'encombrement mémoire requis (mémoire) en donnant le nombre de vecteurs nécessaire à chaque algorithme (il s'agit du nombre total puisqu'il n'est pas nécessaire de stocker tous les vecteurs).

Méthode	T1	T2	T3
Flops	$18k + 10$	$23k + 18$	$25k + 21$
Mémoire	3	4	4

TAB. 2: *Emplacement mémoire général et coût opératoire requis par itération pour la mise en œuvre des algorithmes T1, T2 et T3.*

On voit alors qu'ici le troisième algorithme (T3) est le plus coûteux, quelque soit le critère considéré. Le premier (T1) est le plus avantageux et le deuxième (T2) se situe entre les deux autres. Il demande plus d'opérations que T1, moins que T3 et nécessite plus de mémoire que T1 et autant que T3.

On vérifie à nouveau aisément que ce sont trois algorithmes en $O(n^2)$.

5.2.3 Exemples numériques

Les algorithmes de la sous-section 5.2 ont été programmés en *Matlab* 4.2*c*.1 en double précision. Nous présentons tout d'abord les résultats obtenus pour les algorithmes relatifs aux matrices de Toeplitz. Ceux concernant les matrices de Hankel apparaissent ensuite.

Que ce soit pour les systèmes de Hankel ou de Toeplitz, plusieurs types de matrices sont considérés. Pour chaque type, plusieurs dimensions sont utilisées. Elle sont représentées par un indice (c'est-à-dire, par exemple, que T_{50} représente une matrice de dimension 50).

Pour chaque matrice utilisée, le conditionnement est calculé, à l'aide de la fonction *cond* de *Matlab*, qui représente le rapport de la plus grande valeur singulière sur la plus petite (soit le conditionnement en norme 2).

Dans les deux cas (Hankel et Toeplitz), des matrices de dimension 50, 250 et 1000 sont étudiées afin de voir l'incidence de cette dimension sur chaque algorithme.

Dans tous les exemples, les seconds membres sont toujours choisis au hasard via la fonction *rand* de *Matlab* (leurs éléments sont ainsi compris entre 0 et 1).

Les résultats sont présentés sous forme de tableau où figure la norme euclidienne du résidu obtenu à l'itération n (où n désigne la dimension de la matrice) pour chaque algorithme.

L'accent est mis sur l'algorithme qui présente le meilleur résultat par le fait que le résidu le plus faible figure en caractères gras.

La solution des systèmes considérés est toujours notée $\boldsymbol{x}$.

Matrices de Toeplitz Dans ce paragraphe, nous discutons des résultats obtenus à l'aide des algorithmes T1, T2 et T3 définis à la sous-section 5.2.2. Quatre types de matrices seront étudiés.

Remarque 5.2

Même si pour les algorithmes T2 et T3 un coefficient c_{n-1} est nécessaire, il n'intervient aucunement dans le calcul de P_k (et donc de la solution), $\forall k \leqslant n$. Nous n'étudierons donc pas l'influence de ce coefficient.

– La première matrice $\boldsymbol{T}_n^{(1)}$ considérée est une matrice de Toeplitz dont les éléments sont choisis au hasard entre 0 et 1 (par la fonction *rand* de *Matlab*). Comme une matrice de Toeplitz de dimension n ne dépend que de $2n-1$ coefficients (la première ligne et la première colonne), c désigne ces éléments, c'est-à-dire le vecteur $(c_{-n}, c_{-n+1}, \ldots, c_{n-1})^{\mathrm{T}}$ dans l'écriture de la matrice en (II.47) lorsque n est la dimension utilisée.

Les tableaux 3, 4 et 5 consignent les résultats obtenus.

$n = 50$	T1	T2	T3
$\left\|\boldsymbol{b} - \boldsymbol{T}_{50}^{(1)}\boldsymbol{x}\right\|_2$	$\mathbf{7.34 \times 10^{-13}}$	1.29×10^{-12}	1.10×10^{-12}

TAB. 3: $\boldsymbol{c} = rand(1, 99)$; $\boldsymbol{b} = rand(50, 1)$; $cond(\boldsymbol{T}_{50}^{(1)}) = 1.33 \times 10^3$.

$n = 250$	T1	T2	T3
$\left\|\boldsymbol{b} - \boldsymbol{T}_{250}^{(1)}\boldsymbol{x}\right\|_2$	$\mathbf{2.09 \times 10^{-11}}$	2.60×10^{-11}	3.32×10^{-11}

TAB. 4: $\boldsymbol{c} = rand(1, 499)$; $\boldsymbol{b} = rand(250, 1)$; $cond(\boldsymbol{T}_{250}^{(1)}) = 7.14 \times 10^3$.

$n = 1000$	T1	T2	T3
$\left\|\boldsymbol{b} - \boldsymbol{T}_{1000}^{(1)}\boldsymbol{x}\right\|_2$	8.13×10^{-10}	5.54×10^{-10}	$\mathbf{4.50 \times 10^{-10}}$

TAB. 5: $\boldsymbol{c} = rand(1, 1999)$; $\boldsymbol{b} = rand(1000, 1)$; $cond(\boldsymbol{T}_{1000}^{(1)}) = 1.25 \times 10^4$.

On constate ainsi des résultats corrects en dépit d'un conditionnement assez élevé dans les trois cas. On ne peut pas dire si un algorithme est meilleur qu'un autre vu le peu de différence que l'on observe dans la norme des résidus.

– La deuxième matrice $\boldsymbol{T}_n^{(2)}$ de Toeplitz utilisée est une matrice tridiagonale qui figure dans la *Test Matrix Toolbox* de Higham [45] (elle est nommée *Tridiag*). Son expression générale est la suivante

$$Tridiag : \begin{pmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \end{pmatrix}.$$

Les résultats des tableaux 6, 7 et 8 suivants ont été obtenus pour ce type de matrice.

$n = 50$	T1	T2	T3
$\|\boldsymbol{b} - \boldsymbol{T}_{50}^{(2)}\boldsymbol{x}\|_2$	$\mathbf{6.22 \times 10^{-13}}$	$\mathbf{6.22 \times 10^{-13}}$	2.53×10^{-12}

TAB. 6: $\boldsymbol{T}_{50}^{(2)} = Tridiag(50)$; $\boldsymbol{b} = rand(50, 1)$; $cond(\boldsymbol{T}_{50}^{(2)}) = 1.05 \times 10^3$.

$n = 250$	T1	T2	T3
$\|\boldsymbol{b} - \boldsymbol{T}_{250}^{(2)}\boldsymbol{x}\|_2$	$\mathbf{7.66 \times 10^{-11}}$	1.79×10^{-10}	2.96×10^{-10}

TAB. 7: $\boldsymbol{T}_{250}^{(2)} = Tridiag(250)$; $\boldsymbol{b} = rand(250, 1)$; $cond(\boldsymbol{T}_{250}^{(2)}) = 2.55 \times 10^4$

$n = 1000$	T1	T2	T3
$\|\boldsymbol{b} - \boldsymbol{T}_{1000}^{(2)}\boldsymbol{x}\|_2$	$\mathbf{3.73 \times 10^{-9}}$	1.09×10^{-8}	1.87×10^{-8}

TAB. 8: $\boldsymbol{T}_{1000}^{(2)} = Tridiag(1000)$; $\boldsymbol{b} = rand(1000, 1)$; $cond(\boldsymbol{T}_{1000}^{(2)}) = 1.25 \times 10^4$.

Les mêmes constatations effectuées pour $\boldsymbol{T}^{(1)}$ peuvent être formulées pour ce type de matrice où la différence des normes des résidus n'est pas très importante.

– La troisième matrice de Toeplitz $\boldsymbol{T}_n^{(3)}$ figure encore dans la *Test Matrix Toolbox* de Higham. Il s'agit de la matrice *Parter* dont les éléments sont les suivants

$$Parter : \left(\frac{1}{i-j+\frac{1}{2}}\right)_{\substack{i>0\\ j>0}}.$$

Les résultats des tableaux 9, 10 et 11 furent obtenus.

$n = 50$	T1	T2	T3
$\|\|\boldsymbol{b} - \boldsymbol{T}_{50}^{(3)}\boldsymbol{x}\|\|_2$	$\mathbf{9.85 \times 10^{-15}}$	3.27×10^{-12}	2.47×10^{-10}

TAB. 9: $\boldsymbol{T}_{50}^{(3)} = Parter(50)$; $\boldsymbol{b} = rand(50, 1)$; $cond(\boldsymbol{T}_{50}^{(3)}) = 3.04$.

$n = 250$	T1	T2	T3
$\|\|\boldsymbol{b} - \boldsymbol{T}_{250}^{(3)}\boldsymbol{x}\|\|_2$	$\mathbf{1.25 \times 10^{-13}}$	3.31×10^{-10}	3.38×10^{-10}

TAB. 10: $\boldsymbol{T}_{250}^{(3)} = Parter(250)$; $\boldsymbol{b} = rand(250, 1)$; $cond(\boldsymbol{T}_{250}^{(3)}) = 3.67$.

$n = 1000$	T1	T2	T3
$\|\|\boldsymbol{b} - \boldsymbol{T}_{1000}^{(3)}\boldsymbol{x}\|\|_2$	$\mathbf{4.00 \times 10^{-12}}$	2.41×10^{-8}	2.43×10^{-8}

TAB. 11: $\boldsymbol{T}_{1000}^{(3)} = Parter(1000)$; $\boldsymbol{b} = rand(1000, 1)$; $cond(\boldsymbol{T}_{1000}^{(3)}) = 4.23$.

Paradoxalement, même si la matrice est très bien conditionnée quelle que soit la dimension de cette dernière, la norme des résidus obtenus diminue très vite en fonction de la dimension de ce type de matrice. Ici, par contre, l'algorithme T1 donne de bien meilleurs résultats que les deux autres.

– Enfin, la quatrième et dernière matrice de Toeplitz $\boldsymbol{T}_n^{(4)}$ considérée est la matrice *Pdtoep* de la *Test Matrix Toolbox*. Elle est définie positive et ses éléments sont constitués de somme des $\cos(i-j)$ pour $1 \leqslant i, j \leqslant n$ pondérés par des éléments choisis au hasard via la fonction *rand*.

Ce dernier type de matrice a fourni les résultats des tableaux 12, 13 et 14.

$n = 50$	T1	T2	T3
$\|\|\boldsymbol{b} - \boldsymbol{T}_{50}^{(4)}\boldsymbol{x}\|\|_2$	1.01×10^{-12}	$\mathbf{5.48 \times 10^{-13}}$	1.23×10^{-12}

TAB. 12: $\boldsymbol{T}_{50}^{(4)} = Pdtoep(50)$; $\boldsymbol{b} = rand(50, 1)$; $cond(\boldsymbol{T}_{50}^{(4)}) = 1.63 \times 10^2$.

$n = 250$	T1	T2	T3
$\|\boldsymbol{b} - \boldsymbol{T}_{250}^{(4)}\boldsymbol{x}\|_2$	$\mathbf{6.24 \times 10^{-10}}$	7.42×10^{-10}	6.34×10^{-10}

TAB. 13: $\boldsymbol{T}_{250}^{(4)} = Pdtoep(250)$; $\boldsymbol{b} = rand(250, 1)$; $cond(\boldsymbol{T}_{250}^{(4)}) = 1.32 \times 10^5$.

$n = 250$	T1	T2	T3
$\|\boldsymbol{b} - \boldsymbol{T}_{1000}^{(4)}\boldsymbol{x}\|_2$	9.63×10^{-9}	$\mathbf{9.06 \times 10^{-9}}$	1.07×10^{-8}

TAB. 14: $\boldsymbol{T}_{1000}^{(4)} = Pdtoep(1000)$; $\boldsymbol{b} = rand(1000, 1)$; $cond(\boldsymbol{T}_{1000}^{(4)}) = 8.43 \times 10^4$.

Dans ce dernier exemple, on constate également qu'il n'y a pas une différence significative entre les trois algorithmes, même si le troisième reste toujours en deçà des deux autres.

Matrices de Hankel Nous allons maintenant, à travers deux types de matrices, étudier les différents résultats obtenus avec les algorithmes H1, H2 et H3.

Nous nous intéresserons tout particulièrement à certaines valeurs de c_0 afin de voir si ce coefficient a une grande incidence sur les résultats.

Chaque algorithme est ainsi représenté et le plus faible résidu est mis en évidence par des caractères gras.

– La première matrice $\boldsymbol{H}_n^{(1)}$ est, comme pour le cas Toeplitz, la matrice (de Hankel cette fois-ci) dont les éléments sont choisis au hasard à l'aide de la fonction *rand*. Elle est représentée, si l'on se réfère à (II.38) par le vecteur à $2n - 1$ composantes $\boldsymbol{c} = (c_1, c_2, \ldots, c_{2n-1})^{\mathrm{T}}$.

Les résultats relatifs à ce type de matrice figurent dans les tableaux 15, 16 et 17 de la page suivante.

$n = 50$	H1	H2			H3		
c_0	–	10^{-2}	1	10^2	10^{-2}	1	10^2
$\|\|\boldsymbol{b} - \boldsymbol{H}_{50}^{(1)}\boldsymbol{x}\|\|_2$	7.74×10^{-13}	1.32×10^{-13}	1.25×10^{-13}	$\mathbf{8.93 \times 10^{-14}}$	6.84×10^{-13}	1.87×10^{-11}	1.26×10^{-12}

TAB. 15: $\boldsymbol{c} = rand(1, 99)$; $\boldsymbol{b} = rand(50, 1)$; $cond(\boldsymbol{H}_{50}^{(1)}) = 2.51 \times 10^2$.

$n = 250$	H1	H2			H3		
c_0	–	10^{-2}	1	10^2	10^{-2}	1	10^2
$\|\|\boldsymbol{b} - \boldsymbol{H}_{250}^{(1)}\boldsymbol{x}\|\|_2$	1.20×10^{-9}	8.24×10^{-12}	$\mathbf{7.24 \times 10^{-12}}$	1.16×10^{-11}	2.54×10^{-11}	2.78×10^{-10}	2.16×10^{-9}

TAB. 16: $\boldsymbol{c} = rand(1, 499)$; $\boldsymbol{b} = rand(250, 1)$; $cond(\boldsymbol{H}_{250}^{(1)}) = 9.87 \times 10^2$.

$n = 1000$	H1	H2			H3		
c_0	–	10^{-2}	1	10^2	10^{-2}	1	10^2
$\|\|\boldsymbol{b} - \boldsymbol{H}_{1000}^{(1)}\boldsymbol{x}\|\|_2$	3.37×10^{-8}	1.21×10^{-9}	$\mathbf{3.37 \times 10^{-10}}$	6.46×10^{-10}	5.01×10^{-6}	2.32×10^{-8}	3.33×10^{-7}

TAB. 17: $\boldsymbol{c} = rand(1, 1999)$; $\boldsymbol{b} = rand(1000, 1)$; $cond(\boldsymbol{H}_{1000}^{(1)}) = 2.11 \times 10^4$.

On constate ici que le deuxième algorithme donne de très bons résultats (par rapport aux deux autres) pour la dimension $n = 1000$. En effet, alors que H2 fournit un résidu de 3.37×10^{-10}, pour H3, il n'est que de 5.01×10^{-6} et 3.37×10^{-8} pour H1. On note que l'influence de c_0 est réelle mais non significative.

– La seconde matrice étudiée est *Pdtoep* (la même que dans le cas Toeplitz), sauf bien sûr que les colonnes ont été permutées afin que la matrice obtenue soit de Hankel.
On obtient ainsi les données des tableaux 18, 19 et 20 de la page suivante.

Les constatations sont identiques à celles effectuées dans l'exemple précédent. L'algorithme H2 semble en effet donner des résultats plus intéressants et l'influence du coefficient c_0 ne semble pas vraiment déterminante (sauf peut-être pour H3).

D'une façon générale, pour les algorithmes T1, T2 et T3, il n'apparaît pas que l'un soit vraiment meilleur que les deux autres, même si T1 a donné des résidus légèrement plus faibles dans presque tous les cas (mais il y avait si peu de différence). Ceci peut s'expliquer par le fait que, d'après le Tableau 1, c'est celui qui était le plus avantageux en terme de coût opératoire.

Pour les matrices de Hankel, H2 semble quant à lui un peu plus stable que H1 et nettement plus que H3, notamment lorsque la dimension des matrices considérées croît. Cela peut paraître "normal" compte tenu des conclusions du rapport technique de Gutknecht et al. [43]. Les relations "courtes" qu'utilise H2 doivent, selon ce dernier, être plus stables que celles utilisées par H1. Par contre, on aurait pu penser qu'il en aurait été de même pour H3, ce qui ne semble pas être le cas. Cela peut également s'expliquer par le Tableau 2 puisque c'est H3 qui est le plus exigeant selon les deux critères (nombre d'opérations et encombrement mémoire). H2, de ce point de vue là semble alors un bon compromis.
Ces algorithmes semblent n'être utilisable qu'avec des matrices de dimension (relativement) petite. En effet, déjà avec une dimension 1000, les performances deviennent modestes.

Toutes ces procédures restent encore à être comparées avec les méthodes existantes pour les systèmes de Hankel et Toeplitz et, en particulier, à d'autres procédures basées sur les polynômes orthogonaux [55, 56] (même si celles-ci ont initialement pour but l'inversion des matrices de Hankel et Toeplitz) et avec celles basées sur la méthode de bordage [69, 70]. Ces relations sont également à rapprocher des travaux de Kailath et al. [46] et de Rissanen [57].

$n = 50$	H1	H2			H3		
c_0	–	10^{-2}	1	10^2	10^{-2}	1	10^2
$\left\|\boldsymbol{b} - \boldsymbol{H}_{50}^{(2)}\boldsymbol{x}\right\|_2$	1.91×10^{-12}	$\mathbf{1.35 \times 10^{-13}}$	2.71×10^{-12}	2.22×10^{-13}	2.40×10^{-12}	3.10×10^{-9}	3.17×10^{-12}

TAB. 18: $\boldsymbol{H}_{50}^{(2)} = Pdtoep(50)$ [Hankel] ; $\boldsymbol{b} = rand(50, 1)$; $cond(\boldsymbol{H}_{50}^{(2)}) = 8.74 \times 10^2$.

$n = 250$	H1	H2			H3		
c_0	–	10^{-2}	1	10^2	10^{-2}	1	10^2
$\left\|\boldsymbol{b} - \boldsymbol{H}_{250}^{(2)}\boldsymbol{x}\right\|_2$	3.41×10^{-8}	$\mathbf{2.19 \times 10^{-10}}$	2.25×10^{-10}	2.29×10^{-10}	2.31×10^{-8}	1.04×10^{-8}	2.34×10^{-7}

TAB. 19: $\boldsymbol{H}_{250}^{(2)} = Pdtoep(250)$ [Hankel] ; $\boldsymbol{b} = rand(250, 1)$; $cond(\boldsymbol{H}_{250}^{(2)}) = 5.88 \times 10^3$.

$n = 1000$	H1	H2			H3		
c_0	–	10^{-2}	1	10^2	10^{-2}	1	10^2
$\left\|\boldsymbol{b} - \boldsymbol{H}_{1000}^{(2)}\boldsymbol{x}\right\|_2$	8.02×10^{-8}	$\mathbf{4.51 \times 10^{-9}}$	1.53×10^{-7}	2.35×10^{-8}	7.46×10^{-7}	8.51×10^{-4}	2.91×10^{-6}

TAB. 20: $\boldsymbol{H}_{1000}^{(2)} = Pdtoep(1000)$ [Hankel] ; $\boldsymbol{b} = rand(1000, 1)$; $cond(\boldsymbol{H}_{1000}^{(2)}) = 8.43 \times 10^4$.

Conclusion

On a donc vu, dans cette partie, que la majeure partie des résultats concernant l'interprétation matricielle des polynômes orthogonaux a pu être étendue aux polynômes biorthogonaux.

Malheureusement, l'intérêt des ces résultats ne semble que théorique. En effet, bien que les racines des polynômes biorthogonaux aient été considérées pour certaines relations, aucun nouveau résultat général n'a pu être trouvé, par exemple, quant à la localisation de ces zéros (quelques résultats limités ont déjà été trouvés par Brezinski et al. pour les polynômes *quasi-biorthogonaux* [15] dans le cas où les fonctionnelles $\mathcal{L}_i$ vérifient des hypothèses particulières ainsi que pour les polynômes orthogonaux de dimension $d > 1$).

De tels résultats existent pourtant pour les polynômes orthogonaux [15], même s'ils ne sont que partiels. L'interprétation matricielle des polynômes biorthogonaux, n'a, dans cette optique, rien apporté de nouveau.

Toutefois, d'importants résultats, comme l'extension de l'identité de Christoffel-Darboux ainsi que des résultats sur les propriétés de certaines matrices résolvantes ont pu être obtenus.

De nouvelles relations matricielles liant les familles de polynômes biorthogonaux à gauche et à droite ont également été démontrées.

Enfin, une équivalence a été établie entre calcul des polynômes biorthogonaux et mise en œuvre de la méthode de bordage. Dans ce cadre, des méthodes de résolution des systèmes de Hankel et des systèmes de Toeplitz ont été proposées. Elles nécessitent toutes un nombre d' opérations de l'ordre de $O(n^2)$ où n est la dimension du système considéré. Certains de ces algorithmes semblent être assez efficaces sur des systèmes de dimension.

Troisième partie
Principales méthodes existantes de résolution de systèmes linéaires à seconds membres multiples

Introduction

Le but de cette partie n'est pas de répertorier de façon exhaustive toutes les méthodes de résolution de systèmes linéaires lorsque plusieurs seconds membres sont considérés mais de rappeler celles d'entre elles les plus souvent rencontrées ou les plus récentes, dans la mesure où nous y feront référence, pour certaines, dans la quatrième partie. Nous allons, pour chaque méthode, énoncer les caractéristiques principales de celle-ci.

Dans cette optique, et dans la **première section**, nous rappellerons quelques notions déterminantes pour la compréhension de ces méthodes. Elles ne sont pas nécessairement utiles pour toutes les méthodes mais sont souvent rencontrées.

Ensuite, nous nous consacrerons aux méthodes de résolution proprement dites. Ces dernières sont de deux types essentiellement.

- Les premières généralisent les méthodes de résolution classique en les étendant aux systèmes avec plusieurs seconds membres et sont appelées, en général, méthodes par bloc. Elles seront rappelées dans la **deuxième section**.
- Les secondes sont des méthodes originales mises en œuvre exclusivement dans le cadre de la résolution de problèmes à seconds membres multiples. C'est ainsi que dans la **troisième section** nous considérerons ce type de méthodes.

1 Notations, Définitions, terminologie

Dans cette Section, nous introduirons quelques définitions complémentaires ainsi que certaines terminologies susceptibles de nous éclairer dans les notions utilisées pour la suite de cette partie.

Avant, nous allons définir les conventions et notations utilisées dans la suite de cette partie ainsi que dans la quatrième partie.

- Les lettres minuscules (grecques ou latines) seront des scalaires, généralement complexes.
- Les lettres grecques minuscules en caractères gras désigneront des matrices carrées de dimension s (où s sera le nombre de seconds membres considérés).
- Les lettres latines majuscules en caractères gras seront des matrices carrées de dimension n (où n est la dimension du système à résoudre).
- Les indices seront utilisés pour désigner l'itération (ou la récurrence) considérée.
- Les exposants concerneront les seconds membres de chaque système.
- Les lettres majuscules latines désigneront des polynômes ou des matrices rectangulaires (le contexte sera alors assez explicite pour pouvoir discerner aisément de quel élément il s'agit).
- Les fonctionnelles linéaires reliées à la biorthogonalité seront désignées par des lettres majuscules calligraphiées, tandis que celles directement liées à l'orthogonalité le seront par des lettres latines minuscules. La distinction avec des scalaires sera alors assez claire.

Dans la mesure du possible, nous nous efforcerons de nous tenir à ces conventions d'écriture.

La plupart des méthodes itératives de résolution de systèmes (à second membre unique ou à seconds membres multiples) est basée sur la génération d'un espace de Krylov, que nous allons donc définir ici.

Définition 1.1 – *Sous-espace de Krylov*
Soit $\boldsymbol{A} \in \mathbb{C}^{n\times n}$ *et* $\boldsymbol{v} \in \mathbb{C}^n$. *Le sous espace vectoriel engendré par les vecteurs* $\boldsymbol{v}, \boldsymbol{Av}, \ldots, \boldsymbol{A}^{k-1}\boldsymbol{v}$ *est appelé sous-espace de Krylov d'ordre* k *engendré par* $\boldsymbol{A}$ *et* $\boldsymbol{v}$, *et est noté* $K_k(\boldsymbol{A}, \boldsymbol{v}) = vect(\boldsymbol{v}, \boldsymbol{Av}, \ldots, \boldsymbol{A}^{k-1}\boldsymbol{v})$.

Cette notion a un sens aussi bien si $\boldsymbol{v}$ est un vecteur à n composantes que si $\boldsymbol{v}$ représente une matrice rectangulaire de dimension $n \times s$.

Ainsi, dans le cas où $\boldsymbol{v}$ est un vecteur de $\mathbb{C}^n$, on parlera d'espace de Krylov ou d'espace de Krylov simple si le contexte le nécessite, tandis que si $\boldsymbol{v}$ est

une matrice rectangulaire, on utilisera la terminologie espace de Krylov matriciel.

Qu'elles soient par bloc ou non, les méthodes de résolution de systèmes linéaires à seconds membres multiples considèrent parfois un système central. Nous définissons ici cette notion.

Terminologie

Certaines méthodes de résolution de systèmes linéaires sont basées sur un système central autour duquel s'organise la résolution des autres problèmes. Cette notion est rencontrée dans les ouvrages anglo-saxons sous le nom de **seed system**.

Dans tout ce qui suit dans ce chapitre, nous rappelons que le problème étudié ici est la résolution d'un système linéaire de la forme

$$\boldsymbol{A}X = B \tag{III.1}$$

où $\boldsymbol{A}$ est une matrice carrée inversible de dimension n à coefficients complexes et où X et B sont deux matrices rectangulaires à coefficients complexes de dimension $n \times s$. Le paramètre s désigne donc le nombre de seconds membres considérés. Le but est bien évidemment de trouver la matrice X de dimension $n \times s$ satisfaisant l'équation (III.1) pour une matrice $\boldsymbol{A}$ et un second membre B donnés.

2 Les méthodes par bloc

La majeure partie des méthodes existantes pour la résolution des systèmes linéaires à seconds membres multiples consiste en une généralisation des méthodes existantes pour la résolution des systèmes linéaires "traditionnels" (c'est-à-dire avec un second membre unique). Il s'agit en fait dans ces cas précis de versions par bloc des algorithmes correspondant.

Nous allons donc décrire ici quelques unes des principales méthodes de ce type en nous intéressant tout particulièrement à la méthode introduite par O'Leary [50], que nous utiliserons par la suite. La **première sous-section** sera ainsi consacrée au *Block Bi-CG*.

Dans la **deuxième sous-section**, une version du *Block GCR* sera considérée. Elle est notamment une extension du GCR de Saad que l'on peut trouver dans [58].

Dans la **troisième sous-section** nous rappelons les propriétés essentielles des versions de type *Block QMR*, introduite par Boyse et al. dans [3].

Enfin, dans la **quatrième sous-section** les principales caractéristiques de la version du *Block GMRES*, étudiée par Simoncini et al. dans [62, 64, 65] seront rappelées.

2.1 Le Block Bi-CG

Nous allons tout d'abord décrire dans le détail la méthode du Gradient Biconjugué par bloc avant de considérer les propriétés essentielles de celle-ci. Un algorithme permettant sa mise en œuvre sera également donné dans la mesure où il sera utilisé par la suite.

La méthode du *Block Bi-CG* (ou gradient Biconjugué par bloc) de O'Leary [50] est l'une des méthodes de référence pour la résolution des systèmes linéaires à seconds membres multiples. Il s'agit d'une généralisation de la méthode du Gradient Biconjugué pour les systèmes linéaires non symétriques avec un seul second membre [47] (voir Fletcher [36] pour la forme algorithmique). À ce titre, tout comme dans le cas d'un second membre unique, la matrice considérée peut être quelconque.

Le *Block Bi-CG* est une méthode basée sur la génération d'espaces de Krylov matriciels de plus en plus grand. Elle utilise en outre la transposée de la matrice $\boldsymbol{A}$ et, à chaque itération, il est nécessaire d'inverser deux matrices carrées de

dimension s.

Dans la mesure où cette méthode est utilisée dans le chapitre suivant pour des essais numériques, nous proposons ici un algorithme pour sa mise en œuvre. Cet algorithme est donné dans un cadre général. Certaines restrictions seront faites quant à son application par la suite.

Algorithme Block Bi-CG($\boldsymbol{A}, B, \boldsymbol{M}, X_0, \varepsilon$)

- **Initialisations**

  $R_0 \leftarrow B - \boldsymbol{A}X_0$
  $\bar{R}_0 \leftarrow B - \boldsymbol{A}^{\mathrm{T}}X_0$
  $P_0 \leftarrow \boldsymbol{M}R_0\gamma_0$
  $\bar{P}_0 \leftarrow \boldsymbol{M}^{\mathrm{T}}\bar{R}_0\bar{\gamma}_0$

- **Pour** $k = 0, \ldots$ jusque convergence **Faire**

  $\boldsymbol{\alpha_k} \leftarrow (\bar{P}_k^{\mathrm{T}}\boldsymbol{A}P_k)^{-1}\bar{\gamma}_k^{\mathrm{T}}\bar{R}_k^{\mathrm{T}}\boldsymbol{M}R_k$
  $\bar{\boldsymbol{\alpha}}_{\boldsymbol{k}} \leftarrow (P_k^{\mathrm{T}}\boldsymbol{A}^{\mathrm{T}}\bar{P}_k)^{-1}\gamma_k^{\mathrm{T}}R_k^{\mathrm{T}}\boldsymbol{M}^{\mathrm{T}}\bar{R}_k$
  $\boldsymbol{\beta_k} \leftarrow \gamma_k^{-1}(\bar{R}_k^{\mathrm{T}}\boldsymbol{M}R_k)^{-1}\bar{R}_{k+1}^{\mathrm{T}}\boldsymbol{M}R_{k+1}$
  $\bar{\boldsymbol{\beta}}_{\boldsymbol{k}} \leftarrow \bar{\gamma}_k^{-1}(R_k^{\mathrm{T}}\boldsymbol{M}^{\mathrm{T}}\bar{R}_k)^{-1}R_{k+1}^{\mathrm{T}}\boldsymbol{M}^{\mathrm{T}}\bar{R}_{k+1}$
  $R_{k+1} \leftarrow R_k - \boldsymbol{A}P_k\boldsymbol{\alpha_k}$
  $X_{k+1} \leftarrow X_k + P_k\boldsymbol{\alpha_k}$
  Si $\|R_{k+1}\| \leqslant \varepsilon$ **Alors** Stop.
  $\bar{R}_{k+1} \leftarrow \bar{R}_k - \boldsymbol{A}^{\mathrm{T}}\bar{P}_k\bar{\boldsymbol{\alpha}}_{\boldsymbol{k}}$
  Calcul[1]de γ_{k+1} et de $\bar{\gamma}_{k+1}$
  $P_{k+1} \leftarrow (\boldsymbol{M}R_{k+1} + P_k\boldsymbol{\beta_k})\gamma_{k+1}$
  $\bar{P}_{k+1} \leftarrow (\boldsymbol{M}^{\mathrm{T}}\bar{R}_{k+1} + \bar{P}_k\bar{\boldsymbol{\beta}}_{\boldsymbol{k}})\bar{\gamma}_{k+1}$

  Fin de Pour.

La matrice rectangulaire X_0 correspond à une première estimation de la solution et ε est la précision désirée. La matrice $\boldsymbol{M}$ est une matrice carrée arbitraire de dimension n (en fait il s'agit d'un préconditionneur). Quant aux quantités γ_k et $\bar{\gamma}_k$, il s'agit, pour chaque itération, de deux matrices carrées de dimension et de rang s, qui peuvent être choisies plus ou moins judicieusement[3].

Dans l'algorithme ainsi que dans la suite, $\lfloor n/s \rfloor$ désigne le quotient de la division entière de n par s.

Le Gradient Biconjugué par bloc possède certaines propriétés d'orthogonalité. En particulier, nous avons le

[3]O'Leary préconise de choisir ces deux matrices de telle sorte que les matrices P_k et $\bar{P}_k$ soient orthonormales, en effectuant une décomposition QR ou une orthogonalisation de Gram-Schmidt des matrices $\boldsymbol{M}R_k + P_{k-1}\boldsymbol{\beta}_{k-1}$ et $\boldsymbol{M}^{\mathrm{T}}\bar{R}_k + \bar{P}_{k-1}\bar{\boldsymbol{\beta}}_{k-1}$. Ce processus engendre alors un redémarrage de l'algorithme si une des deux matrices obtenues n'est plus de rang s.

Lemme 2.1 - *(O'Leary [50])*
Les quantités $\bar{R}_k$, R_k, P_k, $\bar{P}_k$ de l'Algorithme 2.1 ainsi que les matrices $\boldsymbol{M}$ et $\boldsymbol{A}$ vérifient les conditions d'orthogonalité matricielle

$$\left.\begin{array}{l} \bar{R}_i^{\mathrm{T}} \boldsymbol{M} R_j = R_i^{\mathrm{T}} \boldsymbol{M}^{\mathrm{T}} \bar{R}_j = 0 \\ \bar{P}_i^{\mathrm{T}} \boldsymbol{A} P_j = P_i^{\mathrm{T}} \boldsymbol{A}^{\mathrm{T}} \bar{P}_j = 0 \end{array}\right\} \textit{pour } j < i.$$

Si l'on considère la forme bilinéaire suivante, définie sur l'espace vectoriel des matrices à coefficients réels de dimension $n \times s$

$$(P, Q)_{\boldsymbol{M}} = P^{\mathrm{T}} \boldsymbol{M} Q$$

où P et Q sont deux matrices de dimension $n \times s$, alors les conditions précédentes peuvent s'écrire

$$\left.\begin{array}{l} (\bar{R}_i, R_j)_{\boldsymbol{M}} = (R_i, \bar{R}_j)_{\boldsymbol{M}^{\mathrm{T}}} = 0 \\ (\bar{P}_i, P_j)_{\boldsymbol{A}} = (P_i, \bar{P}_j)_{\boldsymbol{A}^{\mathrm{T}}} = 0 \end{array}\right\} \text{ pour } j < i, \qquad \text{(III.2)}$$

ce qui justifie la terminologie employée d'orthogonalité matricielle.

Les relations essentielles données en (III.2) permettent notamment d'énoncer le

Théorème 2.1 - *(O'Leary [50])*
La solution exacte X du système linéaire de n équations à $n \times s$ inconnues $\boldsymbol{A}X = B$ est atteinte en au plus $\lfloor n/s \rfloor$ itérations pour la méthode du Block Bi-CG.

Une méthode plus simple mais basée sur une idée analogue peut être obtenue [50] dans le cas où la matrice $\boldsymbol{A}$ considérée est symétrique (éventuellement définie positive). Elle se déduit sans difficulté du Gradient Biconjugué par bloc et est appelée Gradient Conjugué par bloc ou *Block CG*.

2.2 Le Block GCR

La méthode du *Block GCR* ou GCR par bloc est une généralisation du *GCR* de Saad [58]. Il s'agit, tout comme pour le *Block Bi-CG*, d'une méthode basée sur la génération d'espaces de Krylov matriciels de tailles de plus en plus importantes.

Cette méthode requiert également l'inversion, à chaque itération, de matrices carrées de dimension s.

Comme le *Block Bi-CG*, la méthode du *Block GCR* possède des propriétés d'orthogonalité de telle sorte qu'en un nombre d'itérations inférieur ou égal à $\lfloor n/s \rfloor$ la solution des différents systèmes linéaires est obtenue.

L'algorithme du *Block GCR* construit en effet deux suites de matrices P_k et R_k de dimension $n \times s$ de telle sorte que

$$\begin{array}{l}(P_j, P_i)_{\boldsymbol{A}^T\boldsymbol{A}} = 0 \text{ pour } j \neq i \\ (R_j, P_i)_{\boldsymbol{A}} = 0 \text{ pour } i < j\end{array} \tag{III.3}$$

où les formes bilinéaires considérées sont celles de la sous-section 2.1, $R_i = B - \boldsymbol{A}X_i$ représente la matrice résidu obtenue à l'étape i et les matrices P_i sont des matrices auxiliaires utilisées dans l'algorithme *Block GCR*.

Ces propriétés d'orthogonalité ont une conséquence avantageuse certaine : tout comme pour le *Block Bi-CG*, la solution exacte du système linéaire considéré est obtenue en $\lfloor n/s \rfloor$ itérations.

De plus, par construction, le bloc GCR possède des propriétés intéressantes de minimisation des colonnes des résidus R_i sur certains espaces affines.

Néanmoins, cette minimisation a un coût : la mémorisation de toutes les matrices P_i obtenues par les itérations successives, ce qui peut fatalement entraîner un problème d'encombrement mémoire lorsqu'un grand nombre de données est considéré (soit dû à une matrice de dimension importante : cela peut augmenter d'autant le nombre d'itérations à effectuer, soit dû à un nombre de seconds membres conséquent : cela augmente alors la dimension de toutes les matrices à stocker).

Enfin, en considérant les conditions d'orthogonalité données en (III.3), on constate que la transposée $\boldsymbol{A}^T$ de la matrice $\boldsymbol{A}$ est, comme pour le *Block Bi-CG*, utilisée.

2.3 Les méthodes Block QMR

Les méthodes dites *Block QMR* sont une modification de méthodes par bloc existantes. Elles utilisent, de façon générale, la méthode *QMR* présente dans [37]. Ainsi, on peut trouver le *QMR-MBCG* dans [61] qui est une modification du *Block Bi-CG*. Une version *Block QMR* pour les matrices symétriques est également obtenue dans [3] et est basée sur les itérations de Lanczos.

Ainsi, ces méthodes héritent des avantages mais aussi des inconvénients propres à chaque méthode de base.

La méthode mise au point par Boyse, est basée sur la méthode de Lanczos (Gradient conjugué par bloc) et ainsi sur la construction d'espaces de Krylov matriciels. La quasi-minimisation des résidus est obtenue par une décomposition QR. Le comportement de la méthode est meilleur que pour le *Block CG* mais les calculs sont alors beaucoup plus importants (du fait de la décomposition QR).

De plus, pour cette méthode, seulement les matrices symétriques peuvent être considérées, ce qui limite son champ d'application.

La méthode de Simoncini, le *QMR-MBCG* a l'avantage de ne pas se restreindre aux matrices symétriques. Elle utilise une procédure d'orthogonalisation de Gram-Schmidt modifiée ainsi que des rotations de Givens afin d'éviter des décompositions QR pour la minimisation des résidus.

Complexe, cette méthode permet une convergence plus certaine des résidus par rapport au *Block Bi-CG*.

Toutefois, elle utilise encore la transposée de la matrice $\boldsymbol{A}$. Par contre, les matrices $s \times s$ à inverser ici ont une allure beaucoup plus favorable que celles utilisées dans le *Block Bi-CG*. Elles sont en effet triangulaires ou diagonales.

Ainsi, même si le nombre de calculs effectués par *QMR-MBCG* est plus important que pour le *Block Bi-CG*, l'algorithme est plus fiable.

Toutefois, pour ces deux méthodes, une restriction majeure est à apporter : il faut, pour des raisons de coût et d'encombrement, que le nombre de seconds membres soit négligeable devant la dimension de la matrice considérée. Ceci était déjà le cas pour les méthodes du *Block CG* et *Block Bi-CG*.

2.4 Le Block GMRES

La méthode du *Block GMRES*, décrite par Vital [79] est basée sur la méthode d'Arnoldi, une décomposition QR ainsi que sur une orthogonalisation de Gram-Schmidt modifiée, comme pour les méthodes *Block QMR*. Ainsi, un espace de Krylov matriciel est également considéré.

La particularité de cette méthode par rapport aux autres est que l'espace de Krylov utilisé est de taille m fixée dès le départ (c'est déjà le cas pour le *GMRES* lorsqu'un seul second membre est considéré).

Par rapport aux autres méthodes de résolution par bloc, elle possède une propriété de minimisation des résidus. Comme pour le *Block GCR*, cette minimisation s'accompagne d'un besoin de mémoire supérieur, ce qui la rend délicate lorsque les données à traiter sont de grande taille.

C'est pour cela que l'on utilise généralement une version redémarrée de cet algorithme.

3 Autres méthodes

Dans cette Section nous allons considérer des algorithmes qui ont été spécialement conçus pour la résolution des systèmes à seconds membres multiples, en ce sens qu'ils ne sont pas, contrairement aux algorithmes précédents, juste issus de résolution de systèmes linéaires à second membre unique. C'est-à-dire que le fait d'avoir plusieurs seconds membres a été spécialement utilisé.

Ainsi, dans la **première sous-section** nous rappellerons un algorithme élaboré par Simoncini et al. [63] et qui est basé sur une modification et une adaptation du *Block GMRES*.

De même, dans la **seconde sous-section**, deux algorithmes dus à T. F. Chan et al. dans [26] seront considérés.

3.1 Méthode basée sur le GMRES

Récemment, Simoncini et Gallopoulos ont proposé un algorithme hybride basé sur le *Block GMRES*. C'est ainsi que le *MHGMRES* (pour Multiple Hybrid GMRES) est apparu [63].

Cet algorithme utilise lui aussi, comme le *Block GMRES*, un processus d'Arnoldi mais non par bloc cette fois (en utilisant une procédure de Gram-Schmidt modifiée pour l'orthogonalisation). Ainsi un espace de Krylov simple est constitué et les autres résidus sont projetés sur ce dernier.

Un système de moindres carrés est alors résolu via une décomposition QR tandis qu'un système central doit être considéré à chaque itération. Ce dernier peut varier à chaque itération et est choisi selon un critère qui peut être déterminé différemment selon le problème considéré (dans [63], le système central choisi est, selon un procédé heuristique, celui où la norme du résidu est la plus importante).

De plus, un problème de valeurs propres généralisé doit être résolu afin de trouver les racines du polynôme du *GMRES*. Il faut ainsi trouver les valeurs λ qui vérifient

$$\boldsymbol{H}^{\mathrm{T}}\boldsymbol{H}\boldsymbol{z} = \lambda\widetilde{\boldsymbol{H}}^{\mathrm{T}}\boldsymbol{z}.$$

où $\boldsymbol{H}$ est une matrice de Hessenberg supérieure de dimension $(m+1) \times m$ (avec m le paramètre généralement utilisé pour la mise en œuvre du GMRES) et $\widetilde{\boldsymbol{H}} = [\boldsymbol{I}_m, 0]\,\boldsymbol{H}$ et pour un $\boldsymbol{z}$ donné.

Ensuite, la procédure de Richardson est utilisée pour connaître la valeur du polynôme du *GMRES* pour chaque résidu.

Cette méthode présente l'avantage de ne pas utiliser la transposée de la matrice $\boldsymbol{A}$. De plus, l'algorithme *MHGMRES* présente des propriétés de majoration sur les résidus considérés.

En outre, ne s'agissant pas d'une méthode par bloc, le *MHGMRES* ne nécessite pas l'inversion de matrices de dimension s à chaque itération.

Par contre, elle introduit une notion délicate qui est la résolution de problèmes de valeurs propres généralisé (qui n'est pas gênant si la valeur de m est petite).

La version *MHGMRES* donnée dans [63] est une version redémarrée.

3.2 Méthodes basées sur le Gradient Conjugué

Les méthodes originales mises au point par T. F. Chan et al. dans [26] mettent en œuvre deux algorithmes redémarrés. Deux versions sont en effet considérées. L'une utilise un système central tandis que l'autre est une version par bloc qui considère toutefois un système central (mais un système central par bloc cette fois-ci).

Pour le premier algorithme, à chaque redémarrage un système central est choisi parmi les systèmes qui ne sont pas encore résolus.

Dès lors l'algorithme du Gradient Conjugué est appliqué au système central tandis qu'une projection de Galerkin est opérée sur l'espace de Krylov généré par la méthode du Gradient Conjugué sur les autres systèmes et ce jusqu'à résolution du système central considéré.

Ainsi, basée sur la méthode du Gradient Conjugué, uniquement les matrices symétriques définies positives sont considérées.

De plus, les seconds membres considérés par T. F. Chan sont sensés être assez "proches". C'est-à-dire par exemple qu'ils peuvent se représenter par une forme du type

$$B = [\boldsymbol{b}^{(1)}, \ldots, \boldsymbol{b}^{(s)}] \tag{III.4}$$

où $\boldsymbol{b}^{(t)} = b(t)$ avec b fonction vectorielle continue de t.

Cette condition est supposée afin d'entraîner un bon comportement de l'algorithme (même si celui-ci peut être mis en œuvre pour des seconds membres quelconques).

Un Théorème concernant une majoration des résidus est notamment obtenu pour le système central et il en est de même si les autres seconds membres ont l'expression supposée en (III.4). De même, étant basé sur la méthode du Gradient Conjugué, l'algorithme proposé dans [26] est sensé ne redémarrer qu'un nombre fini de fois (en fait il ne peut théoriquement y avoir plus de redémarrages

que le rang de B).

Pour la seconde version de l'algorithme, la version du *Block CG* est utilisée en lieu et place du Gradient Conjugué. Le principe de l'algorithme est le même que dans le cas précédent sauf bien sûr que, cette fois, une projection de Galerkin par bloc est employée.

Des résultats analogues concernant les majorations des résidus sont obtenus pour la méthode par bloc. De plus le nombre de redémarrage est fini car, comme dans le cas de la méthode où le Gradient Conjugué simple est considéré, le nombre de redémarrage ne peut excéder $\lfloor k/s \rfloor - 1$ où $k = \text{rang}(B)$.

Bien sûr, cet algorithme étant basé sur la méthode du *Block CG* et la méthode de Galerkin par bloc, il est nécessaire à chaque itération, et pour chaque second membre, d'inverser une matrice (qui est de dimension de moins en moins grande puisqu'à chaque redémarrage un au moins des systèmes considéré est numériquement résolu).

La méthode par bloc est dans ce cas plus appropriée si les seconds membres ne sont pas "proches", contrairement à la méthode non bloc décrite précédemment.

Il est également à noter qu'aucune indication concernant le choix du système central (pour la méthode non bloc) ou du choix des systèmes centraux (pour la méthode par bloc) n'est fournie dans [26]. En fait T. F. Chan considère, dans chaque cas, les premiers systèmes non résolus.

Conclusion

Cette partie présente donc bon nombre des principales méthodes de résolution de systèmes linéaires à plusieurs seconds membres ainsi que d'autres plus originales, même si d'autres peuvent encore être trouvées [59], [64], [38]. Pour les matrices symétriques, les travaux de Bristeau et al. [21] et de Smith et al. [67] peuvent être considérés.

Les premiers types d'algorithmes (ceux par bloc) présentent l'avantage, en général, de nécessiter peu d'itérations pour obtenir la solution exacte des systèmes étudiés. Cette propriété intéressante s'accompagne souvent d'inconvénients. En effet, soit le nombre de seconds membres considérés doit être très petit (ce que l'on représente souvent par l'inégalité $s \ll n$), soit la place mémoire nécessaire à la mise en œuvre des méthodes décrites est importante du fait d'un stockage de données important, c'est le cas notamment du *Block GCR*.

Les seconds types d'algorithmes ont d'autres caractéristiques intéressantes mais nécessitent parfois d'autres traitements plus complexes que pour les

premiers (décompositions QR, problèmes de valeurs propres généralisés, ...)

Une majeure partie de tous ces algorithmes fait également appel à la transposée de la matrice $\boldsymbol{A}$ à l'origine des systèmes linéaires étudiés. Ce problème est bien entendu évité lorsque la matrice est symétrique. On peut toujours se ramener à un tel cas, soit en considérant la matrice $\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}$, soit en considérant la matrice de dimension $2n$ définie par

$$\begin{pmatrix} 0 & \boldsymbol{A} \\ \boldsymbol{A}^{\mathrm{T}} & 0 \end{pmatrix}$$

Mais, dans le premier cas, cela requiert des opérations supplémentaires et souvent le conditionnement des matrices alors étudiées devient plus important, ce qui rend les algorithmes d'autant plus instables.

Dans le second cas, la taille du problème, qui peut déjà être importante, devient double de celle du problème initial.

On peut donc se demander si l'on ne peut pas trouver une méthode "simple" et efficace qui permettrait de résoudre le problème (III.1), qui n'utiliserait pas la transposée de la matrice $\boldsymbol{A}$ et qui ne serait pas tributaire, pour sa complexité, du nombre de seconds membres en question (c'est-à-dire, par exemple, qu'il ne serait pas nécessaire d'inverser à chaque itération des matrices dont la dimension dépend précisément du nombre de seconds membres). Nous allons tenter de répondre à ce problème dans la Quatrième Partie.

Quatrième partie
Polynômes biorthogonaux, systèmes à seconds membres multiples

Introduction

Cette partie propose de s'intéresser à la résolution de systèmes linéaires de la forme

$$\boldsymbol{A}X = B \tag{IV.1}$$

où $\boldsymbol{A}$ est une matrice inversible non nécessairement symétrique de dimension n et B une matrice rectangulaire $n \times s$ arbitraire. Dans cette partie, s ne doit pas nécessairement être négligeable devant n. Un accent tout particulier sera apporté à l'expression de ce problème en fonction des polynômes biorthogonaux.

On a vu dans la Troisième Partie que nombre de méthodes proposent des versions bloc de méthodes pré-existantes. C'est en effet le cas, rappelons-le, du *Block GCR* (issu dur GCR de Saad [58]), du *Block Bi-CG* de O'Leary (voir [50]), du *Block GMRES* mis au point par Simoncini et al. (voir [62, 64]) ou encore du *Block QMR* étudié également par Simoncini et al. (voir [61]). Toutes ces méthodes, comme nous l'avons vu, nécessitent l'inversion de matrices $s \times s$ à chaque itération. Ainsi, s doit être relativement petit, sinon des difficultés de calcul et des problèmes d'encombrement mémoire risquent d'apparaître.

Certaines méthodes ont vu le jour pour éviter ces problèmes. Généralement, un système central est considéré et les résultats obtenus pour la résolution de ce système sont utilisés pour résoudre les autres systèmes. C'est notamment le cas de l'algorithme proposé par T. F. Chan et al. [26].

D'autres méthodes encore ont récemment été mises au point. Simoncini et al. [63] considèrent une méthode itérative qui nécessite la résolution d'un problème de valeurs propres généralisées. Cette méthode échange les informations obtenues pour la résolution de chaque système afin d'accélérer la convergence.

Pour la résolution de systèmes à second membre unique, beaucoup d'algorithmes sont basés sur la méthode de Lanczos (voir [17] par exemple). Dans cette

partie, on utilise un système central pour résoudre (IV.1). La méthode est basée sur une modification de la méthode de Lanczos d'où l'on tire des algorithmes sans utilisation de la transposée ainsi qu'un algorithme de type BiCGStab (voir [71]). Tous ces algorithmes permettront une interprétation matricielle spécifique ainsi qu'une interprétation polynomiale en termes de polynômes biorthogonaux. Ils peuvent également être partiellement consultés dans [54].

Ainsi la **première section** rappelle brièvement la méthode de Lanczos pour la résolution d'un système linéaire à second membre unique puisque c'est à partir de cette méthode que de nouveaux algorithmes seront construits pour la résolution de systèmes linéaires à seconds membres multiples.

La **deuxième section** décrit les modifications apportées à la méthode de Lanczos pour des seconds membres multiples et considère le BiCGStab avec ces modifications. C'est alors que des considérations en termes de polynômes biorthogonaux seront introduites.

Enfin, dans la **troisième section**, quelques exemples numériques d'application de ces nouveaux algorithmes seront donnés. Ils seront notamment comparés à une version du *Block Bi-CG* donnée dans la Troisième Partie.

1 La méthode de Lanczos et ses mises en œuvre

La méthode considérée dans la Section 2 étant basée sur une modification de la méthode de Lanczos, nous allons, dans une **première sous-section**, rappeler brièvement la méthode de Lanczos et ses caractéristiques.

Dans la **deuxième sous-section**, le lien entre méthode de Lanczos et notion de polynômes orthogonaux formels sera donné.

Dans la **troisième sous-section**, les principales mises en œuvre de la méthode de Lanczos seront considérées, ce qui permettra la justification de la terminologie employée par la suite pour les nouveaux algorithmes.

1.1 La méthode de Lanczos

Rappelons ici en quoi consiste la méthode de Lanczos [48] pour la résolution de systèmes linéaires. Pour cela, considérons le système

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b},$$

où $\boldsymbol{A}$ est une matrice carrée inversible $n \times n$ à coefficients complexes et où $\boldsymbol{x}$ et $\boldsymbol{b} \in \mathbb{C}^n$. La méthode de Lanczos construit deux suites de vecteurs $\{\boldsymbol{r}_k\}_{k\geqslant 0}$ et $\{\boldsymbol{x}_k\}_{k\geqslant 0}$ telles que

$$\boldsymbol{x}_k - \boldsymbol{x}_0 \in K_k(\boldsymbol{A}, \boldsymbol{r}_0) = vect(\boldsymbol{r}_0, \boldsymbol{A}\boldsymbol{r}_0, \ldots, \boldsymbol{A}^{k-1}\boldsymbol{r}_0) \quad \text{(IV.2)}$$

$$\boldsymbol{r}_k = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_k \perp K_k(\boldsymbol{A}^*, \boldsymbol{y}) = vect(\boldsymbol{y}, \boldsymbol{A}^*\boldsymbol{y}, \ldots, \boldsymbol{A}^{*k-1}\boldsymbol{y}) \quad \text{(IV.3)}$$

où $\boldsymbol{A}^*$ désigne la transposée de $\bar{\boldsymbol{A}}$ et où $\boldsymbol{x}_0$ et $\boldsymbol{y}$ sont deux vecteurs (presque) arbitraires. Généralement, on choisit $\boldsymbol{y} = \boldsymbol{r}_0$.

D'après la définition de la suite $\{\boldsymbol{x}_k\}_{k\geqslant 0}$ introduite en (IV.2), on trouve

$$\boldsymbol{x}_k - \boldsymbol{x}_0 = -a_1\boldsymbol{r}_0 - a_2\boldsymbol{A}\boldsymbol{r}_0 - \ldots - a_k\boldsymbol{A}^{k-1}\boldsymbol{r}_0 \quad \text{(IV.4)}$$

où $a_1, \ldots, a_k$ sont des éléments de $\mathbb{C}$.

Et alors, si P_k est le polynôme de degré au plus k défini par

$$P_k(x) = 1 + a_1 x + \ldots + a_k x^k, \quad \text{(IV.5)}$$

on obtient, en multipliant (IV.4) par $-\boldsymbol{A}$ et en ajoutant $\boldsymbol{b}$,

$$\begin{aligned} \boldsymbol{r}_k = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_k &= \boldsymbol{r}_0 + a_1\boldsymbol{A}\boldsymbol{r}_0 + \ldots + a_k\boldsymbol{A}^k\boldsymbol{r}_0 \\ &= P_k(\boldsymbol{A})\boldsymbol{r}_0. \end{aligned}$$

Pour des conditions de régularité (notamment certaines relations de récurrence), on suppose souvent que ce polynôme est de degré k exactement, ce que nous supposerons pour tout k.

Les conditions d'orthogonalité (IV.3) peuvent s'écrire

$$(\boldsymbol{r}_k, \boldsymbol{A}^{*i}\boldsymbol{y}) = (\boldsymbol{A}^i\boldsymbol{r}_k, \boldsymbol{y}) = (\boldsymbol{A}^i P_k(\boldsymbol{A})\boldsymbol{r}_0, \boldsymbol{y}) = 0 \text{ pour } 0 \leqslant i \leqslant k-1.$$

Et les coefficients $a_1, \ldots, a_k$ vérifient alors

$$a_1(\boldsymbol{A}^{i+1}\boldsymbol{r}_0, \boldsymbol{y}) + \ldots + a_k(\boldsymbol{A}^{i+k}\boldsymbol{r}_0, \boldsymbol{y}) = -(\boldsymbol{A}^i\boldsymbol{r}_0, \boldsymbol{y}) \text{ pour } 0 \leqslant i \leqslant k-1. \quad \text{(IV.6)}$$

Le vecteur $(a_1, \ldots, a_k)^{\mathrm{T}}$ est alors la solution du système de Hankel d'ordre k généré par $\Big((\boldsymbol{r}_0, \boldsymbol{A}^*\boldsymbol{y}), \ldots, (\boldsymbol{r}_0, \boldsymbol{A}^{*2k-1}\boldsymbol{y})\Big)^{\mathrm{T}}$ avec le second membre $\Big(-(\boldsymbol{r}_0, \boldsymbol{y}), \ldots, -(\boldsymbol{r}_0, \boldsymbol{A}^{*k-1}\boldsymbol{y})\Big)^{\mathrm{T}}$.

Propriété 1.1

Supposons que les vecteurs $\boldsymbol{y}, \boldsymbol{A}^*\boldsymbol{y}, \ldots, \boldsymbol{A}^{*n-1}\boldsymbol{y}$ *soient linéairement indépendants. Alors*

$$\exists k \leqslant n, \boldsymbol{r}_k = 0.$$

1.2 La méthode de Lanczos et les polynômes orthogonaux formels

Il nous faut maintenant introduire la notion de polynômes orthogonaux formels pleinement étudiés par Draux dans [32] et rappelés dans la première partie. Soit c la fonctionnelle linéaire définie sur l'espace des polynômes à une indéterminée à coefficients complexes par ses moments

$$c(x^i) = c_i = (\boldsymbol{A}^i\boldsymbol{r}_0, \boldsymbol{y}) \text{ pour tout } i \geqslant 0. \quad \text{(IV.7)}$$

Alors, pour tout polynôme P, on a

$$c(P) = (P(\boldsymbol{A})\boldsymbol{r}_0, \boldsymbol{y}). \quad \text{(IV.8)}$$

Ceci est uniquement dû à la linéarité du produit scalaire. Ainsi, les conditions (IV.6) (et donc les conditions d'orthogonalité (IV.3)) deviennent

$$c(x^i P_k) = 0 \text{ pour } 0 \leqslant i \leqslant k-1. \quad \text{(IV.9)}$$

Une famille de polynômes satisfaisant (IV.9) est appelée famille de polynômes orthogonaux formels par rapport à la fonctionnelle linéaire c. Ces polynômes étant

définis à multiplication par une constante près, on les choisit tels que $P_k(0) = 1$, pour que la définition (IV.5) soit cohérente.

De tels polynômes peuvent s'écrire, s'ils existent pour tout k, sous la forme d'un rapport de déterminants

$$P_k(x) = \frac{\begin{vmatrix} 1 & \cdots & x^{k-1} & x^k \\ c_0 & \cdots & c_{k-1} & c_k \\ \vdots & & \vdots & \vdots \\ c_{k-1} & \cdots & c_{2k-2} & c_{2k-1} \end{vmatrix}}{H_k^{(1)}} \tag{IV.10}$$

où

$$H_k^{(1)} = \begin{vmatrix} c_1 & \cdots & c_k \\ \vdots & & \vdots \\ c_k & \cdots & c_{2k-1} \end{vmatrix}.$$

Ainsi, la famille de polynômes $\{P_k\}_{k\geqslant 0}$ existe si et seulement si $H_k^{(1)} \neq 0$ pour tout k. De plus, le polynôme P_k est de degré exactement k si et seulement si

$$H_k^{(0)} = \begin{vmatrix} c_0 & \cdots & c_{k-1} \\ \vdots & & \vdots \\ c_{k-1} & \cdots & c_{2k-2} \end{vmatrix} \neq 0.$$

Dans la suite, nous supposerons que la condition $H_k^{(1)} \neq 0$ est satisfaite pour tout $k > 0$.

On peut introduire la famille de polynômes adjacents $P_k^{(1)}$, orthogonale par rapport à la fonctionnelle linéaire $c^{(1)}$ définie par

$$c^{(1)}(x^i) = c(x^{i+1}) = c_{i+1}.$$

Ces polynômes vérifient

$$c^{(1)}(x^i P_k^{(1)}) = 0 \text{ pour } 0 \leqslant i \leqslant k-1. \tag{IV.11}$$

Ils existent et sont uniques à multiplication par une constante près. On trouve aisément que le polynôme défini par

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

vérifie les conditions d'orthogonalité (IV.11) et qu'en plus il est unitaire. Le dénominateur est ainsi identique à celui qui apparaît dans l'expression sous forme de rapport de déterminants du polynôme P_k donnée en (IV.10). L'existence de P_k et de $P_k^{(1)}$ est alors équivalente, ce qui justifie les normalisations $P_k(0) = 1$ et $P_k^{(1)}$ unitaire.

1.3 Mises en œuvre de la méthode de Lanczos

Les polynômes P_k et $P_k^{(1)}$ précédemment définis vérifient certaines relations de récurrence. De ces diverses relations naîtront diverses possibilités de mise en œuvre la méthode de Lanczos (voir [17]).

Les trois algorithmes principaux sont appelés *Lanczos/Orthodir*, *Lanczos/Orthomin* et *Lanczos/Orthores* et nous allons les rappeler maintenant.

Ces trois algorithmes seront donnés à titre indicatif afin de pouvoir se rendre compte de la différence avec les algorithmes de la Section 2.

1.3.1 Lanczos/Orthodir

Deux premières relations de récurrence concernant les polynômes $P_k^{(1)}$ et les polynômes P_k permettent une première mise en œuvre de la méthode de Lanczos et un premier algorithme.

Comme il est rappelé dans le Théorème I.2 de la Première Partie, les polynômes orthogonaux formels unitaires $P_k^{(1)}$ satisfont une relation de récurrence à trois termes de la forme

$$P_{k+1}^{(1)}(x) = (x - \alpha_k)P_k^{(1)}(x) - \beta_k P_{k-1}^{(1)}(x) \qquad \text{(IV.12)}$$

où $P_{-1}(x) = 0$ et $P_0(x) = 1$.

Nous allons donner une expression différente des coefficients α_k et β_k de celle qui a été donnée dans la Première Partie. Elle nous sera utile dans la suite.

Soit U_i un polynôme de degré exactement i. Par la linéarité de c, les conditions d'orthogonalité (IV.9) peuvent encore s'écrire

$$c(U_i P_k) = 0 \text{ pour } 0 \leqslant i \leqslant k-1. \qquad \text{(IV.13)}$$

De même, on trouvera que (IV.11) peut s'écrire

$$c^{(1)}(U_i P_k^{(1)}) = 0 \text{ pour } 0 \leqslant i \leqslant k-1. \qquad \text{(IV.14)}$$

Ainsi, en multipliant (IV.12) par U_{k-1} et en appliquant $c^{(1)}$, on trouve, en utilisant les conditions d'orthogonalité (IV.14) pour $i = k-1$,

$$\beta_k = \frac{c^{(1)}(xU_{k-1}P_k^{(1)})}{c^{(1)}(U_{k-1}P_{k-1}^{(1)})} \text{ si } k > 0 \text{ et } \beta_0 = 0 \qquad \text{(IV.15)}$$

De même, en multipliant (IV.12) par U_k et en appliquant $c^{(1)}$, on obtient, en utilisant les conditions d'orthogonalité (IV.14) pour $i = k$,

$$\alpha_k = \frac{c^{(1)}(xU_kP_k^{(1)}) - \beta_k c^{(1)}(U_kP_{k-1}^{(1)})}{c^{(1)}(U_kP_k^{(1)})}. \tag{IV.16}$$

On peut également montrer (Brezinski [8]) que les polynômes P_k et les polynômes $P_k^{(1)}$ satisfont une relation de récurrence de la forme

$$P_{k+1}(x) = P_k(x) - \lambda_k x P_k^{(1)}(x) \tag{IV.17}$$

où le coefficient λ_k est donné, en considérant les conditions d'orthogonalité (IV.9) et (IV.11), par

$$\lambda_k = \frac{c(x^kP_k)}{c^{(1)}(x^kP_k^{(1)})}$$

que l'on peut encore exprimer sous la forme

$$\lambda_k = \frac{c(U_kP_k)}{c^{(1)}(U_kP_k^{(1)})} \tag{IV.18}$$

où U_k est un polynôme de degré exactement k (il suffit de multiplier (IV.17) par U_k, d'appliquer c et d'utiliser (IV.13) pour $i = k$).

Enfin, on trouve des expressions plus maniables pour les coefficients β_k, α_k et λ_k en considérant la définition de c introduite en (IV.7), l'égalité (IV.8) et les définitions de ces coefficients respectivement en (IV.15), (IV.16) et (IV.18). On obtient

$$\begin{aligned}
\beta_k &= \frac{(\boldsymbol{A}^2U_{k-1}(\boldsymbol{A})P_k^{(1)}(\boldsymbol{A})\boldsymbol{r}_0, \boldsymbol{y})}{(\boldsymbol{A}U_{k-1}(\boldsymbol{A})P_{k-1}^{(1)}(\boldsymbol{A})\boldsymbol{r}_0, \boldsymbol{y})} \text{ si } k > 0 \\
\alpha_k &= \frac{(\boldsymbol{A}^2U_k(\boldsymbol{A})P_k^{(1)}(\boldsymbol{A})\boldsymbol{r}_0, \boldsymbol{y}) - \beta_k(\boldsymbol{A}U_k(\boldsymbol{A})P_{k-1}^{(1)}(\boldsymbol{A})\boldsymbol{r}_0, \boldsymbol{y})}{(\boldsymbol{A}U_k(\boldsymbol{A})P_k^{(1)}(\boldsymbol{A})\boldsymbol{r}_0, \boldsymbol{y})} \\
\lambda_k &= \frac{(U_k(\boldsymbol{A})P_k(\boldsymbol{A})\boldsymbol{r}_0, \boldsymbol{y})}{(\boldsymbol{A}U_k(\boldsymbol{A})P_k^{(1)}(\boldsymbol{A})\boldsymbol{r}_0, \boldsymbol{y})}.
\end{aligned}$$

Ainsi, en posant $\boldsymbol{q}_k = P_k^{(1)}(\boldsymbol{A})\boldsymbol{r}_0$ et en rappelant que $\boldsymbol{r}_k = P_k(\boldsymbol{A})\boldsymbol{r}_0$, le premier algorithme, que l'on appelle *Lanczos/Orthodir*, s'en suit

Algorithme Lanczos/Orthodir$(\boldsymbol{A}, \boldsymbol{b}, \boldsymbol{x}_0, \boldsymbol{y}, \varepsilon)$

- **Initialisations**

  $\boldsymbol{r}_0 \leftarrow \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_0$

$\boldsymbol{q}_0 = \boldsymbol{r}_0$

- **Itérations**

Pour $k = 0, \ldots$ jusque convergence **Faire**

Si $k = 0$ **Alors**

$\beta_0 = 0$

Sinon

$\beta_k \leftarrow (\boldsymbol{A}^2 U_{k-1}(\boldsymbol{A})\boldsymbol{q}_k, \boldsymbol{y})/(\boldsymbol{A}U_{k-1}(\boldsymbol{A})\boldsymbol{q}_{k-1}, \boldsymbol{y})$

Fin de Si.

$\alpha_k \leftarrow \left[(\boldsymbol{A}^2 U_k(\boldsymbol{A})\boldsymbol{q}_k, \boldsymbol{y}) - \beta_k(\boldsymbol{A}U_k(\boldsymbol{A})\boldsymbol{q}_{k-1}, \boldsymbol{y})\right] / \left[(\boldsymbol{A}U_k(\boldsymbol{A})\boldsymbol{q}_k, \boldsymbol{y})\right]$

$\lambda_k \leftarrow (U_k(\boldsymbol{A})\boldsymbol{r}_k, \boldsymbol{y})/(\boldsymbol{A}U_k(\boldsymbol{A})\boldsymbol{q}_k, \boldsymbol{y})$

$\boldsymbol{q}_{k+1} \leftarrow (\boldsymbol{A} - \alpha_k)\boldsymbol{q}_k - \beta_k \boldsymbol{q}_{k-1}$

$\boldsymbol{r}_{k+1} \leftarrow \boldsymbol{r}_k - \lambda_k \boldsymbol{A}\boldsymbol{q}_k$

$\boldsymbol{x}_{k+1} \leftarrow \boldsymbol{x}_k + \lambda_k \boldsymbol{q}_k$

Si $\|\boldsymbol{r}_{k+1}\| \leqslant \varepsilon$ **Alors** Stop.

Fin de Pour.

Cet algorithme est appelé *Lanczos/Orthodir* si U_k et U_{k-1} sont respectivement $P_k^{(1)}$ et $P_{k-1}^{(1)}$.

1.3.2 Lanczos/Orthomin

Deux autres relations de récurrence permettent d'obtenir une autre mise en œuvre de la méthode de Lanczos.

Il est alors nécessaire d'introduire la famille de polynômes $\{Q_k\}_{k\geqslant 0}$ dont les éléments sont définis par $Q_k = \widetilde{\beta}_k P_k^{(1)}$ où $\widetilde{\beta}_k$ est tel que P_k et Q_k ont le même coefficient du terme de plus haut degré (ils sont tous deux supposés de degré k exactement). On peut facilement montrer que les polynômes Q_k et P_k sont liés par la relation

$$Q_{k+1}(x) = P_{k+1}(x) - \gamma_k Q_k(x) \tag{IV.19}$$

où γ_k est donné par

$$\gamma_k = \frac{c(xU_k P_{k+1})}{c^{(1)}(U_k Q_k)} = \frac{(\boldsymbol{A}U_k(\boldsymbol{A})P_{k+1}(\boldsymbol{A})\boldsymbol{r}_0, \boldsymbol{y})}{(\boldsymbol{A}U_k(\boldsymbol{A})Q_k(\boldsymbol{A})\boldsymbol{r}_0, \boldsymbol{y})}$$

où U_k est un polynôme de degré exactement k. En effet, Q_k vérifie, par définition, les mêmes conditions d'orthogonalité que $P_k^{(1)}$. On multiplie (IV.19) par xU_k et on applique c. Les conditions d'orthogonalité donnent la première expression de γ_k. La seconde est obtenue en considérant (IV.8).

Ainsi, si l'on pose $\widetilde{\boldsymbol{q}}_k = Q_k(\boldsymbol{A})\boldsymbol{r}_0$, on obtient une mise en œuvre de la méthode de Lanczos dont l'algorithme s'appelle *Lanczos/Orthomin.*

Algorithme Lanczos/Orthomin($\boldsymbol{A}, \boldsymbol{b}, \boldsymbol{x}_0, \boldsymbol{y}, \varepsilon$)

- **Initialisations**

  $\boldsymbol{r}_0 \leftarrow \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_0$

  $\boldsymbol{q}_0 = \boldsymbol{r}_0$

- **Itérations**

  Pour $k = 1, \ldots$ jusque convergence **Faire**

  $\gamma_k \leftarrow (\boldsymbol{A}U_k(\boldsymbol{A})\boldsymbol{r}_{k+1}, \boldsymbol{y})/(\boldsymbol{A}U_k(\boldsymbol{A})\widetilde{\boldsymbol{q}}_k, \boldsymbol{y})$

  $\lambda_k \leftarrow (U_k(\boldsymbol{A})\boldsymbol{r}_k, \boldsymbol{y})/(\boldsymbol{A}U_k(\boldsymbol{A})\widetilde{\boldsymbol{q}}_k, \boldsymbol{y})$

  $\boldsymbol{r}_{k+1} \leftarrow \boldsymbol{r}_k - \lambda_k \boldsymbol{A}\widetilde{\boldsymbol{q}}_k$

  $\boldsymbol{x}_{k+1} \leftarrow \boldsymbol{x}_k + \lambda_k \widetilde{\boldsymbol{q}}_k$

  Si $\|\boldsymbol{r}_{k+1}\| \leqslant \varepsilon$ **Alors** Stop.

  $\widetilde{\boldsymbol{q}}_{k+1} \leftarrow \boldsymbol{r}_{k+1} - \gamma_k \widetilde{\boldsymbol{q}}_k$

  Fin de Pour.

Cet algorithme est appelé *Lanczos/Orthomin* lorsque $U_k \equiv P_k$.

1.3.3 Lanczos/Orthores

Enfin, en utilisant uniquement le fait que les polynômes P_k sont eux-aussi des polynômes orthogonaux, on obtient une dernière mise en œuvre de la méthode de Lanczos, que l'on appelle *Lanczos/Orthores.*

Comme les polynômes P_k sont des polynômes orthogonaux, non unitaires, par rapport à c, s'ils existent pour tout k et s'ils sont tous de degré exactement k, ils satisfont une relation de récurrence à trois termes qui peut toujours s'écrire sous la forme

$$P_{k+1}(x) = \alpha_k \left[(x + \beta_k)P_k(x) - \gamma_k P_{k-1}(x)\right] \tag{IV.20}$$

où les coefficients α_k et β_k sont donnés par

$$\gamma_k = \frac{c(xU_{k-1}P_k)}{c(U_{k-1}P_{k-1})} = \frac{(\boldsymbol{A}U_{k-1}(\boldsymbol{A})P_k(\boldsymbol{A})\boldsymbol{r}_0, \boldsymbol{y})}{(U_{k-1}(\boldsymbol{A})P_{k-1}(\boldsymbol{A})\boldsymbol{r}_0, \boldsymbol{y})}$$

$$\beta_k = \frac{\gamma_k c(U_k P_{k-1}) - c(xU_k P_k)}{c(U_k P_k)} = \frac{\gamma_k (U_k(\boldsymbol{A})P_{k-1}(\boldsymbol{A})\boldsymbol{r}_0, \boldsymbol{y}) - (\boldsymbol{A}U_k(\boldsymbol{A})P_k(\boldsymbol{A})\boldsymbol{r}_0, \boldsymbol{y})}{(U_k(\boldsymbol{A})P_k(\boldsymbol{A})\boldsymbol{r}_0, \boldsymbol{y})}.$$

La première expression des coefficients est obtenue en considérant les conditions d'orthogonalité que vérifient les polynômes P_k. On multiplie ainsi (IV.20) par U_{k-1} et l'on applique c, ce qui donne γ_k (on peut diviser par α_k car les polynômes P_k sont de degré k exactement, ce qui impose $\alpha_k \neq 0$). De même, en multipliant (IV.20) par U_k et en appliquant c à nouveau, β_k est obtenu. Les secondes expressions sont à nouveau trouvées à l'aide de (IV.8).

Enfin, la normalisation $P_k(0) = 1$ nous donne

$$\alpha_k = \frac{1}{\beta_k - \gamma_k}.$$

Ainsi, on obtient l'algorithme *Lanczos/Orthores.*

Algorithme Lanczos/Orthores$(\boldsymbol{A}, \boldsymbol{b}, \boldsymbol{x}_0, \boldsymbol{y}, \varepsilon)$

- **Initialisations**

  $\boldsymbol{r}_0 \leftarrow \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_0$

- **Itérations**

  Pour $k = 1, \ldots$ jusque convergence **Faire**

  Si $k = 0$ **Alors**

  $\gamma_0 = 0$

  Sinon

  $\gamma_k \leftarrow (\boldsymbol{A}U_{k-1}(\boldsymbol{A})\boldsymbol{r}_k, \boldsymbol{y})/(U_{k-1}(\boldsymbol{A})\boldsymbol{r}_{k-1}, \boldsymbol{y})$

  Fin de Si.

  $\beta_k \leftarrow [\gamma_k(U_k(\boldsymbol{A})\boldsymbol{r}_{k-1}, \boldsymbol{y}) - (\boldsymbol{A}U_k(\boldsymbol{A})\boldsymbol{r}_k, \boldsymbol{y})]\,/(U_k(\boldsymbol{A})\boldsymbol{r}_k, \boldsymbol{y})$

  $\alpha_k \leftarrow 1/(\beta_k - \gamma_k)$

  $\boldsymbol{r}_{k+1} \leftarrow \alpha_k\,[(\boldsymbol{A} + \beta_k)\boldsymbol{r}_k - \gamma_k\boldsymbol{r}_{k-1}]$

  $\boldsymbol{x}_{k+1} \leftarrow \alpha_k\,[\beta_k\boldsymbol{x}_k - \gamma_k\boldsymbol{x}_{k-1} - \boldsymbol{r}_k]$

  Si $\|\boldsymbol{r}_{k+1}\| \leqslant \varepsilon$ **Alors** Stop.

  Fin de Pour.

Cet algorithme est appelé *Lanczos/Orthores* si l'on choisit $U_k \equiv P_k$ et $U_{k-1} \equiv P_{k-1}$.

Il est connu que les algorithmes utilisant les espaces de Krylov sont plus stables lorsque l'on considère des relations de récurrence "courtes" (comme c'est le cas pour *Lanczos/Orthomin*) que lorsque l'on considère des relations de récurrence "longues" (comme c'est le cas pour *Lanczos/Orthodir* et *Lanczos/Orthores*). Ceci est en effet un résultat du rapport technique de Gutknecht et al. [43].

2 Considération de plusieurs seconds membres

La méthode de Lanczos et ses mises en œuvre décrites dans la Section 1 étant très souvent étudiées, on peut se demander dans quelle mesure on peut trouver, sur les mêmes bases, une méthode de résolution de systèmes linéaires lorsque plusieurs seconds membres sont considérés.

Ainsi, dans la **première sous-section** une modification sera apportée à la méthode de Lanczos dans cette optique. Tandis que la méthode de Lanczos pour la résolution de systèmes linéaires à second membre unique utilise exclusivement des polynômes orthogonaux, une modification de cette dernière considérera des polynômes biorthogonaux pour la résolution de systèmes linéaires avec plusieurs seconds membres.

Dans la **deuxième sous-section** une propriété essentielle de cette nouvelle méthode sera considérée.

Enfin, dans la **troisième sous-section**, les diverses mises en œuvre de cette méthode seront considérées, par analogie avec les mises en œuvre de la méthode de Lanczos. Une modification du BiCGStab de Van Der Vorst [71] sera également proposée.

2.1 Description de la méthode

Nous allons décrire la méthode issue d'une modification de la méthode de Lanczos pour le cas d'un second membre unique. Un système central sera alors considéré où certaines informations seront recherchées pour la résolution des autres systèmes linéaires.

Pour cela, considérons désormais le système linéaire

$$AX = B$$

où $B = \left[\boldsymbol{b}^{(1)}, \ldots, \boldsymbol{b}^{(s)}\right]$ est une matrice de dimension $n \times s$. Chaque élément $\boldsymbol{b}^{(j)}$ est ainsi un vecteur de dimension n. Considérons alors les $2s$ suites $\left\{\boldsymbol{x}_k^{(j)}\right\}_{\substack{1 \leqslant j \leqslant s \\ k \geqslant 0}}$ et $\left\{\boldsymbol{r}_k^{(j)}\right\}_{\substack{1 \leqslant j \leqslant s \\ k \geqslant 0}}$ définies par

$$\boldsymbol{x}_k^{(j)} - \boldsymbol{x}_0^{(j)} \in K_k(\boldsymbol{A}, \boldsymbol{z}) \tag{IV.21}$$

$$\boldsymbol{r}_k^{(j)} = \boldsymbol{b}^{(j)} - \boldsymbol{A}\boldsymbol{x}_k^{(j)} \perp K_k(\boldsymbol{A}^*, \boldsymbol{y}). \tag{IV.22}$$

où $\boldsymbol{z}$, $\boldsymbol{y}$ et $\boldsymbol{x}_0^{(j)}$ $(j = 1, \ldots, s)$ sont choisis (presque) arbitrairement.

Alors, on trouve d'après (IV.21)

$$\boldsymbol{x}_k^{(j)} = \boldsymbol{x}_0^{(j)} + a_1^{(j)}\boldsymbol{z} + \ldots + a_k^{(j)}\boldsymbol{A}^{k-1}\boldsymbol{z} \text{ pour } k > 0 \qquad \text{(IV.23)}$$

où les coefficients $a_1^{(j)}, \ldots, a_k^{(j)}$ sont des éléments de $\mathbb{C}$.

Et ainsi, pour $k > 0$, en multipliant (IV.23) par $-\boldsymbol{A}$ et en retranchant $\boldsymbol{b}^{(j)}$, on obtient

$$\begin{aligned} \boldsymbol{r}_k^{(j)} &= \boldsymbol{b}^{(j)} - \boldsymbol{A}\boldsymbol{x}_k^{(j)} \\ &= \boldsymbol{b}^{(j)} - \boldsymbol{A}\boldsymbol{x}_0^{(j)} - a_1^{(j)}\boldsymbol{A}\boldsymbol{z} - \ldots - a_k^{(j)}\boldsymbol{A}^k\boldsymbol{z} \\ &= \boldsymbol{r}_0^{(j)} - \boldsymbol{A}\Phi_{k-1}^{(j)}(\boldsymbol{A})\boldsymbol{z} \end{aligned}$$

où $\Phi_{k-1}^{(j)}$ est le polynôme de degré au plus $k-1$ défini par

$$\Phi_{k-1}^{(j)}(x) = a_1^{(j)} + a_2^{(j)}x + \ldots + a_k^{(j)}x^{k-1}.$$

Les conditions d'orthogonalité (IV.22) peuvent alors s'écrire, pour tout $k > 0$,

$$a_1^{(j)}(\boldsymbol{A}^{i+1}\boldsymbol{z}, \boldsymbol{y}) + \ldots + a_k^{(j)}(\boldsymbol{A}^{i+k}\boldsymbol{z}, \boldsymbol{y}) = (\boldsymbol{A}^i\boldsymbol{r}_0^{(j)}, \boldsymbol{y}) \text{ pour } \begin{cases} i = 0, \ldots, k-1 \\ j = 1, \ldots, s. \end{cases}$$

Les vecteurs $\left(a_1^{(j)}, \ldots, a_k^{(j)}\right)^{\mathrm{T}}$ sont les solutions des systèmes de Hankel générés par $\left((\boldsymbol{A}\boldsymbol{z}, \boldsymbol{y}), \ldots, (\boldsymbol{A}^{2k-1}\boldsymbol{z}, \boldsymbol{y})\right)^{\mathrm{T}}$ avec les seconds membres $\left((\boldsymbol{r}_0^{(j)}, \boldsymbol{y}), \ldots, (\boldsymbol{A}^{k-1}\boldsymbol{r}_0^{(j)}, \boldsymbol{y})\right)^{\mathrm{T}}$ pour $j = 1, \ldots, s$.

Remarque 2.1

Cette interprétation matricielle est similaire à celle obtenue dans la méthode de Lanczos originale, à un détail près. La matrice de Hankel formée ici **ne dépend pas** *d'un éventuel $\boldsymbol{r}_0$ (donc du second membre) comme c'était le cas dans la Section 1.*

2.2 Un processus fini

Comme dans la méthode de Lanczos originale, et sous une certaine condition, on montre que la nouvelle méthode donne le résultat exact théorique en un nombre fini d'itérations.

Proposition 2.1

Supposons que les vecteurs $\boldsymbol{y}, \boldsymbol{A}^*\boldsymbol{y}, \ldots, \boldsymbol{A}^{*n-1}\boldsymbol{y}$ soient linéairement indépendants. Alors,

$$\exists k \leqslant n,\ \boldsymbol{r}_k^{(j)} = 0 \text{ pour } j = 1, \ldots, s.$$

Preuve :

Par les conditions d'orthogonalité, on obtient $\boldsymbol{r}_n^{(j)} \perp (\boldsymbol{y}, \boldsymbol{A}^*\boldsymbol{y}, \ldots, \boldsymbol{A}^{*n-1}\boldsymbol{y})$ qui sont n vecteurs linéairement indépendants. Or, $\boldsymbol{r}_n^{(j)}$ étant un vecteur à n composantes, il est nécessairement nul. Le résultat est alors évident et les solutions exactes sont donc obtenues en au plus n itérations.

■

La condition de la Proposition précédente n'est naturellement vérifiable, tout comme pour la méthode de Lanczos où un seul second membre est considéré, qu'en calcul exact.

2.3 Mises en œuvre de la nouvelle méthode

Explorons maintenant les possibilités de mise en œuvre de cette méthode. Les fonctionnelles linéaires utilisées sont d'abord définies. Ensuite, nous construisons les suites qui mènent aux solutions des systèmes.

De là, nous tirerons les diverses manières de mettre en œuvre la méthode proposée à la sous-section 2.1.

2.3.1 Fonctionnelles linéaires associées – Expression polynomiale

Considérons les $n \times s$ fonctionnelles linéaires définies sur l'espace des polynômes par

$$\mathcal{L}_i^{(j)}(x^m) = (\boldsymbol{A}^{i+m}\boldsymbol{z}, \boldsymbol{y}) \text{ si } m > 0 \tag{IV.24}$$

$$\mathcal{L}_i^{(j)}(1) = -(\boldsymbol{A}^i \boldsymbol{r}_0^{(j)}, \boldsymbol{y}) \tag{IV.25}$$

pour $1 \leqslant i \leqslant n$ et $1 \leqslant j \leqslant s$.

De façon analogue à la méthode de Lanczos originale, définissons la fonctionnelle linéaire $c^{(1)}$ par

$$c^{(1)}(x^i) = c_{i+1} = (\boldsymbol{A}^{i+1}\boldsymbol{z}, \boldsymbol{y}) \text{ pour } i \geqslant 0.$$

On trouve aisément la

Proposition 2.2
Les fonctionnelles $\mathcal{L}_i^{(j)}$ et $c^{(1)}$ sont liées par

$$c^{(1)}(x^i) = \mathcal{L}_l^{(j)}(x^{i-m+1}) \text{ pour } i - m + 1 > 0.$$

Preuve :

Par définition, $\mathcal{L}_m^{(j)}(x^{i-m+1}) = (\boldsymbol{A}^{m+i-m+1}\boldsymbol{z}, \boldsymbol{y})$ puisque $i - m + 1 \neq 0$. Ainsi, on a $\mathcal{L}_m^{(j)}(x^{i-m+1}) = (\boldsymbol{A}^{i+1}\boldsymbol{z}, \boldsymbol{y}) = c^{(1)}(x^i)$.

■

Soient $\widetilde{P}_k$ les polynômes orthogonaux formels par rapport à $c^{(1)}$. Ils satisfont

$$c^{(1)}(x^i\widetilde{P}_k) = 0 \text{ pour } 0 \leqslant i \leqslant k-1. \tag{IV.26}$$

On les choisit unitaires et on rappelle alors un résultat de la Première Partie.

Propriété 2.1 *(Draux [32])*
Les polynômes orthogonaux formels unitaires $\widetilde{P}_k$ par rapport à $c^{(1)}$ satisfont la relation de récurrence à trois termes

$$\widetilde{P}_{k+1}(x) = (x - \alpha_k)\widetilde{P}_k(x) - \beta_k\widetilde{P}_{k-1}(x) \tag{IV.27}$$

avec $\widetilde{P}_0(x) = 1$, $\widetilde{P}_{-1}(x) = 0$ et

$$\begin{aligned}
\beta_0 &= 0 \\
\beta_k &= \frac{c^{(1)}(x^k\widetilde{P}_k)}{c^{(1)}(x^{k-1}\widetilde{P}_{k-1})} \text{ pour } k > 0 && \text{(IV.28)} \\
\alpha_k &= \frac{c^{(1)}(x^{k+1}\widetilde{P}_k) - \beta_k c^{(1)}(x^k\widetilde{P}_{k-1})}{c^{(1)}(x^k\widetilde{P}_k)} \text{ pour } k \geqslant 0. && \text{(IV.29)}
\end{aligned}$$

L'expression des coefficients est due à l'orthogonalité par rapport à $c^{(1)}$ des polynômes $\widetilde{P}_k$.

En multipliant chaque côté de (IV.27) par un polynôme U_{k-1} de degré exactement $k-1$, en appliquant $c^{(1)}$ et en utilisant les propriétés d'orthogonalité (IV.26) que satisfait $\widetilde{P}_k$, on trouve

$$\beta_k = \frac{c^{(1)}(xU_{k-1}\widetilde{P}_k)}{c^{(1)}(U_{k-1}\widetilde{P}_{k-1})} \text{ pour } k > 0. \tag{IV.30}$$

De plus, en multipliant chaque côté de (IV.27) par un polynôme U_k de degré exactement k, en appliquant $c^{(1)}$ et en utilisant les propriétés d'orthogonalité (IV.26) que vérifie $\widetilde{P}_k$, on obtient également

$$\alpha_k = \frac{c^{(1)}(xU_k\widetilde{P}_k) - \beta_k c^{(1)}(U_k\widetilde{P}_{k-1})}{c^{(1)}(U_k\widetilde{P}_k)} \text{ pour } k \geqslant 0. \tag{IV.31}$$

Maintenant, en posant

$$P_k^{(j)}(x) = 1 + a_1^{(j)}x + \ldots + a_k^{(j)}x^k \tag{IV.32}$$

pour $1 \leqslant j \leqslant s$, on a

$$\begin{aligned} &P_k^{(j)}(0) = 1 \\ &\mathcal{L}_i^{(j)}(P_k^{(j)}) = 0 \text{ pour } i = 0, \ldots, k-1. \end{aligned} \tag{IV.33}$$

En effet, on remplace x par 0 dans (IV.32), ce qui conduit à la première égalité. La linéarité des $\mathcal{L}_i^{(j)}$ et leur définition en (IV.24) et (IV.25) fournissent la seconde.

Ainsi, les polynômes $P_k^{(j)}$ sont les polynômes biorthogonaux introduits par Brezinski dans [8]. Ils sont de degré au plus k et si $P_{k+1}^{(j)}$ est de degré exactement $k+1$, il satisfait la relation

$$P_{k+1}^{(j)}(x) = P_k^{(j)}(x) + \lambda_k^{(j)} x \widetilde{P}_k(x) \tag{IV.34}$$

où $\widetilde{P}_k$ est le polynôme unitaire de degré k tel que

$$\mathcal{L}_i^{(j)}(x\widetilde{P}_k) = 0 \text{ pour } 0 \leqslant i \leqslant k-1 \tag{IV.35}$$

et où

$$\lambda_k^{(j)} = -\frac{\mathcal{L}_k^{(j)}(P_k^{(j)})}{\mathcal{L}_k^{(j)}(x\widetilde{P}_k)}. \tag{IV.36}$$

Or, d'après la Proposition 2.2, on trouve $\mathcal{L}_i^{(j)}(x\widetilde{P}_k) = \mathcal{L}_i^{(j)}(x^{i-i+1}\widetilde{P}_k) = c^{(1)}(x^i\widetilde{P}_k)$ et ainsi (IV.35) devient

$$c^{(1)}(x^i\widetilde{P}_k) = 0 \text{ pour } 0 \leqslant i \leqslant k-1.$$

Les polynômes $\widetilde{P}_k$ sont ainsi les polynômes orthogonaux unitaires par rapport à $c^{(1)}$ et (IV.36) devient

$$\lambda_k^{(j)} = -\frac{\mathcal{L}_k^{(j)}(P_k^{(j)})}{c^{(1)}(x^k\widetilde{P}_k)}. \tag{IV.37}$$

Les polynômes $P_k^{(j)}$ peuvent être écrits, par définition

$$P_k^{(j)}(x) = 1 + x\Phi_{k-1}^{(j)}(x)$$

et ainsi, les polynômes $\Phi_k^{(j)}$ vérifient, si l'on utilise (IV.34),

$$\Phi_k^{(j)}(x) = \Phi_{k-1}^{(j)}(x) + \lambda_k^{(j)}\widetilde{P}_k(x).$$

Et, comme $\boldsymbol{x}_k^{(j)} = \boldsymbol{x}_0^{(j)} + \Phi_{k-1}^{(j)}(\boldsymbol{A})\boldsymbol{z}$ et $\boldsymbol{r}_k^{(j)} = \boldsymbol{b}^{(j)} - \boldsymbol{A}\boldsymbol{x}_k^{(j)}$, on obtient, en posant $\boldsymbol{q}_k = \widetilde{P}_k(\boldsymbol{A})\boldsymbol{z}$

$$\begin{aligned} \boldsymbol{x}_{k+1}^{(j)} &= \boldsymbol{x}_0^{(j)} + \Phi_k^{(j)}(\boldsymbol{A})\boldsymbol{z} = \boldsymbol{x}_0^{(j)} + \Phi_{k-1}^{(j)}(\boldsymbol{A})\boldsymbol{z} + \lambda_k^{(j)}\widetilde{P}_k(\boldsymbol{A})\boldsymbol{z} = \boldsymbol{x}_k^{(j)} + \lambda_k^{(j)}\boldsymbol{q}_k \\ \boldsymbol{r}_{k+1}^{(j)} &= \boldsymbol{b}^{(j)} - \boldsymbol{A}\boldsymbol{x}_{k+1}^{(j)} = \boldsymbol{b}^{(j)} - \boldsymbol{A}(\boldsymbol{x}_k^{(j)} + \lambda_k^{(j)}\boldsymbol{q}_k) = \boldsymbol{r}_k^{(j)} - \lambda_k^{(j)}\boldsymbol{A}\boldsymbol{q}_k. \end{aligned} \quad \text{(IV.38)}$$

Remarque 2.2

Contrairement à la méthode de Lanczos originale, les polynômes $P_k^{(j)}$ ne sont pas, en général, des polynômes orthogonaux. En effet, on n'a pas, dans le cas général, $\mathcal{L}_i^{(j)}(1) = \mathcal{L}_{i-1}^{(j)}(x)$. De plus, nous n'avons pas, comme dans le cas de la méthode de Lanczos avec un second membre unique, une relation du type $\boldsymbol{r}_k^{(j)} = P_k^{(j)}(\boldsymbol{A})\boldsymbol{r}_0^{(j)}$.

Les polynômes $\widetilde{P}_k$ et $P_k^{(j)}$ peuvent s'exprimer sous forme de rapports de déterminants. Pour $\widetilde{P}_k$, il suffit de consulter la Première Partie.

Le polynôme $P_k^{(j)}$, quant à lui, peut s'exprimer par

$$P_k^{(j)}(x) = \frac{\begin{vmatrix} 1 & x & \cdots & x^k \\ -(\boldsymbol{r}_0^{(j)}, \boldsymbol{y}) & c_1 & \cdots & c_k \\ \vdots & \vdots & & \vdots \\ -(\boldsymbol{A}^{k-1}\boldsymbol{r}_0^{(j)}, \boldsymbol{y}) & c_k & \cdots & c_{2k-1} \end{vmatrix}}{\begin{vmatrix} c_1 & \cdots & c_k \\ \vdots & & \vdots \\ c_k & \cdots & c_{2k-1} \end{vmatrix}}.$$

En remplaçant x par 0, on obtient de suite $P_k^{(j)}(0) = 1$. De plus, en appliquant $\mathcal{L}_i^{(j)}$ à la première ligne du déterminant du numérateur et en utilisant la Proposition 2.2, on obtient les conditions d'orthogonalité (IV.33).

Ainsi, les polynômes $\widetilde{P}_k$ et $P_k^{(j)}$ existent si et seulement si $H_k^{(1)} \neq 0$ (on rappelle que $H_k^{(1)}$ est le déterminant du dénominateur) et les polynômes $P_k^{(j)}$ sont des polynômes de degré k exactement si et seulement si

$$H_{k,j}^{(0)} = \begin{vmatrix} (\boldsymbol{r}_0^{(j)}, \boldsymbol{y}) & c_1 & \cdots & c_{k-1} \\ \vdots & \vdots & & \vdots \\ (\boldsymbol{r}_0^{(j)}, \boldsymbol{A}^{*k-1}\boldsymbol{y}) & c_k & \cdots & c_{2k-2} \end{vmatrix} \neq 0.$$

Si les polynômes $\widetilde{P}_k$ et $P_k^{(j)}$ n'existent pas, un problème se pose. Une telle situation est connue pour le cas d'un second membre unique. Ce problème a été résolu pour les polynômes orthogonaux par une technique de sauts développée par Brezinski

et al. [13] et plus récemment dans [14] mais aussi par Brezinski et al. [18, 19, 20]. D'autres techniques existent également pour traiter ce problème, voir [42] et [1].

Ces techniques devraient pouvoir s'appliquer ici car les polynômes $\widetilde{P}_k$ sont des polynômes orthogonaux. De plus, les dénominateurs qui interviennent dans le calcul des polynômes biorthogonaux $P_k^{(j)}$ ne dépendent justement que de ces polynômes orthogonaux. Cela reste à étudier. Nous allons, dans la suite, supposer que tous ces polynômes existent.

2.3.2 Analogie avec Lanczos/Orthodir

Donnons maintenant des expressions utiles des coefficients α_k, β_k et $\lambda_k^{(j)}$. Nous verrons qu'alors un algorithme semblable à *Lanczos/Orthodir* peut être obtenu pour la résolution de systèmes linéaires à seconds membres multiples.

Par définition des fonctionnelles $\mathcal{L}_i^{(j)}$ et $c^{(1)}$ et en utilisant leur linéarité, on montre facilement, en posant $\boldsymbol{q}_k = \widetilde{P}_k(\boldsymbol{A})\boldsymbol{z}$, que

$$\begin{aligned} c^{(1)}(x^i\widetilde{P}_k) &= (\boldsymbol{A}^{i+1}\widetilde{P}_k(\boldsymbol{A})\boldsymbol{z},\boldsymbol{y}) \\ &= (\boldsymbol{A}^{i+1}\boldsymbol{q}_k,\boldsymbol{y}) \qquad \text{(IV.39)} \\ \mathcal{L}_i^{(j)}(P_k^{(j)}) &= -(\boldsymbol{A}^i\boldsymbol{r}_0^{(j)},\boldsymbol{y}) + (\boldsymbol{A}^{i+1}\Phi_{k-1}^{(j)}(\boldsymbol{A})\boldsymbol{z},\boldsymbol{y}) \text{ si } k>0 \\ &= -(\boldsymbol{A}^i\boldsymbol{r}_k^{(j)},\boldsymbol{y}). \qquad \text{(IV.40)} \end{aligned}$$

Il s'agit en effet de considérer la linéarité des fonctionnelles utilisées ainsi que leur définition. On note que, d'après (IV.25), la dernière formulation reste valable si $k = 0$. Ces relations permettent ainsi de déduire une expression utilisable des divers coefficients.

Proposition 2.3

Soient U_k et $V_k^{(j)}$ des polynômes arbitraires de degré k exactement. Alors les coefficients $\lambda_k^{(j)}$, α_k et β_k introduits précédemment peuvent s'écrire

$$\begin{aligned} \lambda_k^{(j)} &= \frac{(\boldsymbol{A}^k\boldsymbol{r}_k^{(j)},\boldsymbol{y})}{(\boldsymbol{A}^{k+1}\boldsymbol{q}_k,\boldsymbol{y})} = \frac{(V_k^{(j)}(\boldsymbol{A})\boldsymbol{r}_k^{(j)},\boldsymbol{y})}{(\boldsymbol{A}V_k^{(j)}(\boldsymbol{A})\boldsymbol{q}_k,\boldsymbol{y})} \qquad \text{(IV.41)} \\ \beta_k &= \frac{(\boldsymbol{A}^{k+1}\boldsymbol{q}_k,\boldsymbol{y})}{(\boldsymbol{A}^k\boldsymbol{q}_{k-1},\boldsymbol{y})} = \frac{(\boldsymbol{A}^2U_{k-1}(\boldsymbol{A})\boldsymbol{q}_k,\boldsymbol{y})}{(\boldsymbol{A}U_{k-1}(\boldsymbol{A})\boldsymbol{q}_{k-1},\boldsymbol{y})} \\ \alpha_k &= \frac{(\boldsymbol{A}^{k+2}\boldsymbol{q}_k,\boldsymbol{y}) - \beta_k(\boldsymbol{A}^{k+1}\boldsymbol{q}_{k-1},\boldsymbol{y})}{(\boldsymbol{A}^{k+1}\boldsymbol{q}_k,\boldsymbol{y})} \\ &= \frac{(\boldsymbol{A}^2U_k(\boldsymbol{A})\boldsymbol{q}_k,\boldsymbol{y}) - \beta_k(\boldsymbol{A}U_k(\boldsymbol{A})\boldsymbol{q}_{k-1},\boldsymbol{y})}{(\boldsymbol{A}U_k(\boldsymbol{A})\boldsymbol{q}_k,\boldsymbol{y})}. \end{aligned}$$

Preuve :

La première expression de β_k est obtenue en considérant (IV.28) et (IV.39). La seconde est obtenue à l'aide de (IV.30) et de (IV.39).

De même, la première expression du coefficient α_k est une conséquence de (IV.29) et de (IV.39) tandis que la seconde vient de (IV.31) et de (IV.39).

Quant à la première expression du coefficient $\lambda_k^{(j)}$, on l'obtient à l'aide de (IV.37), de (IV.39) et de (IV.40). Les polynômes $P_k^{(j)}$ ne faisant pas partie d'une famille de polynômes orthogonaux formels, la seconde égalité est plus délicate. Il faut en effet considérer le fait (que l'on montre aisément par récurrence) que $\boldsymbol{q}_k \perp \boldsymbol{A}^{*i}\boldsymbol{y}$ pour $1 \leqslant i \leqslant k$. Ainsi, si le coefficient de plus haut degré de $V_k^{(j)}$ est $v_k^{(j)}$, alors $(\boldsymbol{A}V_k^{(j)}(\boldsymbol{A})\boldsymbol{q}_k, \boldsymbol{y}) = v_k^{(j)}(\boldsymbol{A}^{k+1}\boldsymbol{q}_k, \boldsymbol{y})$. De même, comme $\boldsymbol{r}_k^{(j)} \perp \boldsymbol{A}^{*i}$ pour $0 \leqslant i \leqslant k-1$, on aura $(V_k^{(j)}(\boldsymbol{A})\boldsymbol{r}_k^{(j)}, \boldsymbol{y}) = v_k^{(j)}(\boldsymbol{A}^k\boldsymbol{r}_k^{(j)}, \boldsymbol{y})$. Et le résultat s'en suit (par simplification des $v_k^{(j)}$).

■

Remarque 2.3

L'expression des coefficients α_k et β_k est strictement analogue à l'expression que l'on peut trouver dans le cadre de la résolution de système linéaire à second membre unique pour Lanczos/Orthodir (puisque les polynômes $\widetilde{P}_k$ sont des polynômes orthogonaux formels). La différence majeure réside dans les coefficients $\lambda_k^{(j)}$ car les polynômes $P_k^{(j)}$ sont des polynômes biorthogonaux et généralement non orthogonaux formels.

Pour des raisons évidentes d'encombrement mémoire, il paraît nécessaire que les polynômes $V_k^{(j)}$ qui interviennent dans (IV.41) ne dépendent pas de j.

Ainsi, en posant $U_k(x) = x^k$ et $V_k^{(j)}(x) = x^k$ pour tout j, on obtient immédiatement l'algorithme Multiple Lanczos/Orthodir suivant qui permet la résolution de systèmes linéaires à seconds membres multiples, similaire à *Lanczos/Orthodir* (pour un système à second membre unique).

Cet algorithme sera appelé M–Lanczos/Orthodir en référence à Multiple Lanczos/Orthodir. Les données B et X_0 représentent respectivement la matrice second membre et une première estimation de la solution $(X_0 = x_0^{(1)}, \ldots, x_0^{(s)})$. De même, $R_0 = (r_0^{(1)}, \ldots, r_0^{(s)})$. Ces notations seront identiques dans les algorithmes suivants.

Algorithme M–Lanczos/Orthodir($\boldsymbol{A}, B, X_0, , \boldsymbol{y}, \boldsymbol{z}, \varepsilon$)

- **Initialisations**

$\boldsymbol{q}_0 = \boldsymbol{z}$
$\boldsymbol{y}_0 = \boldsymbol{y}$
$R_0 \leftarrow B - \boldsymbol{A}X_0$

- **Itérations**

Pour $k = 0, \ldots$ jusque convergence **Faire**
$\bar{\boldsymbol{q}}_k \leftarrow \boldsymbol{A}\boldsymbol{q}_k$
$d_k \leftarrow (\bar{\boldsymbol{q}}_k, \boldsymbol{y}_k)$
Pour $j = 1, \ldots, s$ **Faire**
$\lambda_k^{(j)} \leftarrow (\boldsymbol{r}_k^{(j)}, \boldsymbol{y}_k)/d_k$
$\boldsymbol{r}_{k+1}^{(j)} \leftarrow \boldsymbol{r}_k^{(j)} - \lambda_k^{(j)}\bar{\boldsymbol{q}}_k$
$\boldsymbol{x}_{k+1}^{(j)} \leftarrow \boldsymbol{x}_k^{(j)} + \lambda_k^{(j)}\boldsymbol{q}_k$
Fin de Pour.
Si $\max\limits_{1 \leqslant j \leqslant s} \left\| \boldsymbol{r}_{k+1}^{(j)} \right\| \leqslant \varepsilon$ **Alors** Stop.
$\beta_k \leftarrow d_k/d_{k-1}$
$\boldsymbol{y}_{k+1} \leftarrow \boldsymbol{A}^*\boldsymbol{y}_k$
$\alpha_k \leftarrow \left[(\bar{\boldsymbol{q}}_k, \boldsymbol{y}_{k+1}) - \beta_k(\bar{\boldsymbol{q}}_{k-1}, \boldsymbol{y}_k)\right]/d_k$
$\boldsymbol{q}_{k+1} \leftarrow \bar{\boldsymbol{q}}_k - \alpha_k\boldsymbol{q}_k - \beta_k\boldsymbol{q}_{k-1}$
Fin de Pour.

Le problème de cet algorithme, comme l'a justement fait remarquer Simoncini (communication personnelle), est qu'il nécessite le calcul des puissances itérées de $\boldsymbol{A}^*$ appliquées à un vecteur $\boldsymbol{y}$, ce qui est connu pour être numériquement instable. Il apparaît donc nécessaire de faire un choix plus judicieux pour les polynômes U_k et $V_k^{(j)}$ afin d'éviter un tel inconvénient.

Ainsi, on peut choisir $V_k^{(j)} \equiv U_k$ pour tout $1 \leqslant j \leqslant s$. On a alors besoin, d'après les expressions de la Proposition 2.3, uniquement des quantités $U_k(\boldsymbol{A})\boldsymbol{q}_k$, $U_{k-1}(\boldsymbol{A})\boldsymbol{q}_k$, $U_k(\boldsymbol{A})\boldsymbol{q}_{k-1}$ et $U_k(\boldsymbol{A})\boldsymbol{r}_k^{(j)}$.

Un choix "naturel" pour U_k est $U_k \equiv \widetilde{P}_k$ (puisqu'alors, par définition, on aura $U_{k-1}(\boldsymbol{A})\boldsymbol{q}_k = \widetilde{P}_{k-1}(\boldsymbol{A})\widetilde{P}_k(\boldsymbol{A})\boldsymbol{z} = \widetilde{P}_k(\boldsymbol{A})\widetilde{P}_{k-1}(\boldsymbol{A})\boldsymbol{z} = U_k(\boldsymbol{A})\boldsymbol{q}_{k-1}$).

Donc il nous faut calculer les vecteurs $\widetilde{P}_k(\boldsymbol{A})\boldsymbol{q}_k$, $\widetilde{P}_{k-1}(\boldsymbol{A})\boldsymbol{q}_k$ et $\widetilde{P}_k(\boldsymbol{A})\boldsymbol{r}_k^{(j)}$ en utilisant les relations de récurrence des polynômes $\widetilde{P}_k$ et $P_k^{(j)}$.

On utilise la technique trouvée dans [16] qui consiste à multiplier les relations de récurrences que vérifient les différents polynômes afin d'obtenir les relations nécessaires à l'expression des vecteurs voulus.

Proposition 2.4
En posant $\bar{\boldsymbol{r}}_k^{(j)} = \widetilde{P}_k(\boldsymbol{A})\boldsymbol{r}_k^{(j)}$, $\bar{\boldsymbol{q}}_k = \widetilde{P}_k(\boldsymbol{A})\boldsymbol{q}_k$, $\widehat{\boldsymbol{q}}_k = \widetilde{P}_{k-1}(\boldsymbol{A})\boldsymbol{q}_k$ *et* $\widehat{\boldsymbol{r}}_k^{(j)} =$

$\widetilde{P}_{k-1}(\boldsymbol{A})\boldsymbol{r}_k^{(j)}$, *on a*

$$\begin{aligned}
\bar{\boldsymbol{q}}_{k+1} &= (\boldsymbol{A}-\alpha_k)^2\bar{\boldsymbol{q}}_k - 2\beta_k(\boldsymbol{A}-\alpha_k)\widehat{\boldsymbol{q}}_k + \beta^2\bar{\boldsymbol{q}}_{k-1} \\
\widehat{\boldsymbol{q}}_{k+1} &= (\boldsymbol{A}-\alpha_k)\bar{\boldsymbol{q}}_k - \beta_k\widehat{\boldsymbol{q}}_k \\
\widehat{\boldsymbol{r}}_{k+1}^{(j)} &= \bar{\boldsymbol{r}}_k^{(j)} - \lambda_k^{(j)}\boldsymbol{A}\bar{\boldsymbol{q}}_k \\
\bar{\boldsymbol{r}}_{k+1}^{(j)} &= (\boldsymbol{A}-\alpha_k)\widehat{\boldsymbol{r}}_{k+1}^{(j)} - \beta_k\widehat{\boldsymbol{r}}_k^{(j)} + \lambda_k^{(j)}\beta_k\boldsymbol{A}\widehat{\boldsymbol{q}}_k \\
&= (\boldsymbol{A}-\alpha_k)\bar{\boldsymbol{r}}_k^{(j)} - \beta_k\widehat{\boldsymbol{r}}_k^{(j)} - \lambda_k^{(j)}\boldsymbol{A}\widehat{\boldsymbol{q}}_{k+1}
\end{aligned}$$

avec

$$\begin{aligned}
\lambda_k^{(j)} &= \frac{(\bar{\boldsymbol{r}}_k^{(j)}, \boldsymbol{y})}{(\boldsymbol{A}\bar{\boldsymbol{q}}_k, \boldsymbol{y})} \\
\beta_k &= \frac{(\boldsymbol{A}^2\widehat{\boldsymbol{q}}_k, \boldsymbol{y})}{(\boldsymbol{A}\bar{\boldsymbol{q}}_{k-1}, \boldsymbol{y})} \\
\alpha_k &= \frac{(\boldsymbol{A}^2\bar{\boldsymbol{q}}_k, \boldsymbol{y}) - \beta_k(\boldsymbol{A}\widehat{\boldsymbol{q}}_k, \boldsymbol{y})}{(\boldsymbol{A}\bar{\boldsymbol{q}}_k, \boldsymbol{y})}.
\end{aligned}$$

Preuve :

Par définition de $\boldsymbol{q}_{k+1}$, on a $\bar{\boldsymbol{q}}_{k+1} = \widetilde{P}_{k+1}(\boldsymbol{A})\widetilde{P}_{k+1}(\boldsymbol{A})\boldsymbol{z}$. Ainsi, en utilisant la relation de récurrence à trois termes (IV.27), on obtient $\bar{\boldsymbol{q}}_{k+1} = \left[(\boldsymbol{A}-\alpha_k)\widetilde{P}_k(\boldsymbol{A}) - \beta_k\widetilde{P}_{k-1}(\boldsymbol{A})\right]^2\boldsymbol{z}$. En développant le carré et en remarquant que $\widetilde{P}_{k-1}(\boldsymbol{A})\widetilde{P}_k(\boldsymbol{A})\boldsymbol{z} = \widehat{\boldsymbol{q}}_k$, l'expression obtenue de $\bar{\boldsymbol{q}}_{k+1}$ apparaît.

Par définition de $\boldsymbol{q}_{k+1}$, on a $\widehat{\boldsymbol{q}}_{k+1} = \widetilde{P}_k(\boldsymbol{A})\widetilde{P}_{k+1}(\boldsymbol{A})\boldsymbol{z}$. En utilisant à nouveau la relation de récurrence (IV.27), on obtient $\widehat{\boldsymbol{q}}_{k+1} = \widetilde{P}_k(\boldsymbol{A})\left[(\boldsymbol{A}-\alpha_k)\widetilde{P}_k(\boldsymbol{A}) - \beta_k\widetilde{P}_{k-1}(\boldsymbol{A})\right]\boldsymbol{z}$, d'où l'expression de $\widehat{\boldsymbol{q}}_{k+1}$.

Par la relation que vérifie $\boldsymbol{r}_{k+1}^{(j)}$ en (IV.38), on a $\widehat{\boldsymbol{r}}_{k+1}^{(j)} = \widetilde{P}_k(\boldsymbol{A})\left[\boldsymbol{r}_k^{(j)} - \lambda_k^{(j)}\boldsymbol{A}\boldsymbol{q}_k\right]$. D'où l'expression obtenue de $\widehat{\boldsymbol{r}}_{k+1}^{(j)}$.

Par définition de $\bar{\boldsymbol{r}}_{k+1}^{(j)}$, on a $\bar{\boldsymbol{r}}_{k+1}^{(j)} = \widetilde{P}_{k+1}(\boldsymbol{A})\boldsymbol{r}_{k+1}^{(j)}$. En utilisant la relation de récurrence (IV.27), on trouve $\bar{\boldsymbol{r}}_{k+1}^{(j)} = \left[(\boldsymbol{A}-\alpha_k)\widetilde{P}_k(\boldsymbol{A}) - \beta_k\widetilde{P}_{k-1}(\boldsymbol{A})\right]\boldsymbol{r}_{k+1}^{(j)}$. En utilisant la relation (IV.38), on obtient $\bar{\boldsymbol{r}}_{k+1}^{(j)} = \left[(\boldsymbol{A}-\alpha_k)\widetilde{P}_k(\boldsymbol{A})\right]\boldsymbol{r}_{k+1}^{(j)} - \beta_k\widetilde{P}_{k-1}(\boldsymbol{A})\left[\boldsymbol{r}_k^{(j)} - \lambda_k^{(j)}\boldsymbol{A}\boldsymbol{q}_k\right]$, ce qui nous donne la première expression obtenue

de $\bar{\boldsymbol{r}}_{k+1}^{(j)}$. La seconde expression est obtenue en utilisant la première ainsi que l'expression de $\widehat{\boldsymbol{r}}_{k+1}^{(j)}$.

Enfin, en remplaçant U_k et $V_k^{(j)}$ dans la Proposition 2.3 par $\widetilde{P}_k$, le résultat est évident pour les coefficients α_k, β_k et $\lambda_k^{(j)}$.

■

Remarque 2.4
Par rapport à ce que l'on peut trouver dans [16], on n'a pas la même souplesse pour évaluer les vecteurs de la Proposition précédente dans la mesure où $P_k^{(j)}(\boldsymbol{A})\boldsymbol{z} \neq r_k^{(j)}$ en général. Or une relation de ce type est utilisée dans [16].

De la Proposition 2.4 on obtient un algorithme sans utilisation de la transposée, que l'on nomme TFM–Lanczos/Orthodir (pour Transpose–Free Multiple Lanczos/Orthodir).

Algorithme TFM–Lanczos/Orthodir$(\boldsymbol{A}, B, X_0, \boldsymbol{y}, \boldsymbol{z}, \varepsilon)$

- **Initialisations**

$\bar{\boldsymbol{q}}_0 = \boldsymbol{q}_0 = \boldsymbol{z}$
$R_0 \leftarrow B - \boldsymbol{A}X_0$
$\bar{R}_0 = R_0$

- **Itérations**

Pour $k = 0, \ldots$ jusque convergence **Faire**
$\dot{\boldsymbol{q}}_k \leftarrow \boldsymbol{A}\widehat{\boldsymbol{q}}_k$
$\widetilde{\boldsymbol{q}}_k \leftarrow \boldsymbol{A}\bar{\boldsymbol{q}}_k$
$\ddot{\boldsymbol{q}}_k \leftarrow \boldsymbol{A}\widetilde{\boldsymbol{q}}_k$
$d_k \leftarrow (\widetilde{\boldsymbol{q}}_k, \boldsymbol{y})$
$\beta_k \leftarrow (\boldsymbol{A}\dot{\boldsymbol{q}}_k, \boldsymbol{y})/d_{k-1}$
$\alpha_k \leftarrow [(\ddot{\boldsymbol{q}}_k, \boldsymbol{y}) - \beta_k(\dot{\boldsymbol{q}}_k, \boldsymbol{y})]/d_k$
Pour $j = 1, \ldots, s$ **Faire**
$\lambda_k^{(j)} \leftarrow (\bar{\boldsymbol{r}}_k^{(j)}, \boldsymbol{y})/d_k$
$\widehat{\boldsymbol{r}}_{k+1}^{(j)} \leftarrow \bar{\boldsymbol{r}}_k^{(j)} - \lambda_k^{(j)}\widetilde{\boldsymbol{q}}_k$
$\bar{\boldsymbol{r}}_{k+1}^{(j)} \leftarrow (\boldsymbol{A} - \alpha_k)\widehat{\boldsymbol{r}}_{k+1}^{(j)} - \beta_k\widehat{\boldsymbol{r}}_k^{(j)} + \lambda_k^{(j)}\beta_k\dot{\boldsymbol{q}}_k$
$\boldsymbol{x}_{k+1}^{(j)} \leftarrow \boldsymbol{x}_k^{(j)} + \lambda_k^{(j)}\boldsymbol{q}_k$
$\boldsymbol{r}_{k+1}^{(j)} \leftarrow \boldsymbol{r}_k^{(j)} - \lambda_k^{(j)}\boldsymbol{A}\boldsymbol{q}_k$
Fin de Pour.
Si $\max\limits_{1 \leqslant j \leqslant s} \left\| \boldsymbol{r}_{k+1}^{(j)} \right\| \leqslant \varepsilon$ **Alors** Stop.
$\bar{\boldsymbol{q}}_{k+1} \leftarrow \ddot{\boldsymbol{q}}_k - 2\alpha_k\widetilde{\boldsymbol{q}}_k + \alpha_k^2\bar{\boldsymbol{q}}_k - 2\beta_k(\dot{\boldsymbol{q}}_k - \alpha_k\widehat{\boldsymbol{q}}_k) + \beta_k^2\bar{\boldsymbol{q}}_{k-1}$

$\widehat{\boldsymbol{q}}_{k+1} \leftarrow \widetilde{\boldsymbol{q}}_k - \alpha_k \bar{\boldsymbol{q}}_k - \beta_k \widehat{\boldsymbol{q}}_k$
$\boldsymbol{q}_{k+1} \leftarrow (\boldsymbol{A} - \alpha_k)\boldsymbol{q}_k - \beta_k \boldsymbol{q}_{k-1}$
Fin de Pour.

Il faut remarquer que les vecteurs $\boldsymbol{x}_k^{(j)}$ sont, par définition, identiques à ceux évalués dans M–Lanczos/Orthodir. Cet algorithme présente donc un intérêt limité dans la mesure où le résidu relatif à $\boldsymbol{x}_k^{(j)}$ ne peut être calculé récursivement. Les vecteurs intermédiaires sont uniquement utilisés pour le calcul des produits scalaires !

2.3.3 Analogie avec Lanczos/Orthomin

L'algorithme *Lanczos/Orthomin*, rappelé en sous-section 1.3.2, utilise certains polynômes P_{k+1} et Q_k pour calculer le polynôme Q_{k+1}. Nous allons faire de même pour obtenir un algorithme analogue pour la résolution des systèmes linéaires à seconds membres multiples. Voyons alors à quoi peuvent correspondre les polynômes P_k et Q_k.

Ainsi, on pose $\boldsymbol{b}^{(0)} = \boldsymbol{A}\boldsymbol{x}_0^{(0)} - \boldsymbol{z}$ et l'on considère les deux suites $\left\{\boldsymbol{x}_k^{(0)}\right\}_{k>0}$ et $\left\{\boldsymbol{r}_k^{(0)}\right\}_{k>0}$ définies par

$$\begin{aligned} \boldsymbol{x}_k^{(0)} - \boldsymbol{x}_0^{(0)} &\in K_k(\boldsymbol{A}, \boldsymbol{z}) \\ \boldsymbol{r}_k^{(0)} = \boldsymbol{b}^{(0)} - \boldsymbol{A}\boldsymbol{x}_k^{(0)} = \boldsymbol{A}\boldsymbol{x}_0^{(0)} - \boldsymbol{z} - \boldsymbol{A}\boldsymbol{x}_k^{(0)} &\perp K_k(\boldsymbol{A}^*, \boldsymbol{y}) \end{aligned}$$

Alors

$$\boldsymbol{x}_k^{(0)} = \boldsymbol{x}_0^{(0)} + a_1^{(0)}\boldsymbol{z} + \ldots + a_k^{(0)}\boldsymbol{A}^{k-1}\boldsymbol{z} \text{ pour } k > 0$$

où $a_1^{(0)}, \ldots, a_k^{(0)}$ sont des complexes. Alors, en posant

$$P_k^{(0)}(x) = 1 + a_1^{(0)}x + \ldots + a_k^{(0)}x^k,$$

on obtient $\boldsymbol{r}_k^{(0)} = \boldsymbol{b}^{(0)} - \boldsymbol{A}\boldsymbol{x}_k^{(0)} = \boldsymbol{A}\boldsymbol{x}_0^{(0)} - \boldsymbol{z} - \boldsymbol{A}\boldsymbol{x}_k^{(0)} = \boldsymbol{A}\boldsymbol{x}_0^0 - z - \boldsymbol{A}\left(\boldsymbol{x}_0^{(0)} + a_1^{(0)}\boldsymbol{z} + \ldots + a_k^{(0)}\boldsymbol{A}^{k-1}\boldsymbol{z}\right) = -\boldsymbol{z} - a_1^{(0)}\boldsymbol{A}\boldsymbol{z} - \ldots - a_k^{(0)}\boldsymbol{A}^k\boldsymbol{z} = -P_k^{(0)}(\boldsymbol{A})\boldsymbol{z}.$

On peut poser $\Phi_{k-1}^{(0)}(x) = a_1^{(0)} + a_2^{(0)}x + \cdots + a_k^{(0)}x^{k-1}$ comme précédemment.

On introduit la fonctionnelle linéaire c définie par

$$c(x^i) = (\boldsymbol{A}^i\boldsymbol{z}, \boldsymbol{y}) \text{ pour } i \geqslant 0.$$

Alors, par définition de c et par linéarité,

$$c(x^i P_k^{(0)}) = (\boldsymbol{A}^i P_k^{(0)}(\boldsymbol{A})\boldsymbol{z}, \boldsymbol{y}) = 0 \text{ pour } 0 \leqslant i \leqslant k-1$$

et la famille $\left\{P_k^{(0)}\right\}$ est donc la famille de polynômes orthogonaux formels par rapport à c.

Les fonctionnelles linéaires c et $c^{(1)}$ seront ainsi liées par la relation

$$c^{(1)}(x^i) = c(x^{i+1}) = (\boldsymbol{A}^{i+1}\boldsymbol{z}, \boldsymbol{y}).$$

Considérons alors le polynôme $Q_k = \widetilde{\beta}_k \widetilde{P}_k$ où $\widetilde{\beta}_k$ est choisi tel que Q_k et $P_k^{(0)}$ ont le même coefficient de terme de plus haut degré (on suppose ces deux polynômes de degré k exactement).

Les conditions d'orthogonalité que vérifient les polynômes $P_k^{(0)}$ et $\widetilde{P}_k$ et les normalisations utilisées nous donnent alors les relations de récurrence

$$P_{k+1}^{(0)}(x) = P_k^{(0)}(x) + \lambda_k^{(0)} x Q_k \tag{IV.42}$$

$$Q_{k+1}(x) = P_{k+1}^{(0)}(x) + \gamma_k Q_k(x). \tag{IV.43}$$

Par orthogonalité, les coefficients $\lambda_k^{(0)}$ et γ_k sont déterminés par

$$\lambda_k^{(0)} = -\frac{c(x^k P_k^{(0)})}{c^{(1)}(x^k Q_k)}$$

$$\gamma_k = -\frac{c(x^{k+1} P_{k+1}^{(0)})}{c^{(1)}(x^k Q_k)}.$$

Les relations (IV.42) et (IV.43) sont donc analogues à celles obtenues dans le cadre de l'algorithme *Lanczos/Orthomin*. Encore faut-il pouvoir les appliquer à la résolution de systèmes linéaires à plusieurs seconds membres.

En posant $\boldsymbol{q}_k = Q_k(\boldsymbol{A})\boldsymbol{z}$, on obtient, d'après (IV.42) et (IV.43), $\boldsymbol{q}_{k+1} = \gamma_k \boldsymbol{q}_k - \boldsymbol{r}_{k+1}^{(0)}$ et $\boldsymbol{r}_{k+1}^{(0)} = \boldsymbol{r}_k^{(0)} - \lambda_k^{(0)} \boldsymbol{A}\boldsymbol{q}_k$.

Proposition 2.5

Si U_k et $V_k^{(0)}$ désignent des polynômes de degré k exactement, les coefficients γ_k et $\lambda_k^{(0)}$ peuvent s'écrire

$$\gamma_k = \frac{(\boldsymbol{A}^{k+1}\boldsymbol{r}_{k+1}^{(0)}, \boldsymbol{y})}{(\boldsymbol{A}^{k+1}\boldsymbol{q}_k, \boldsymbol{y})} = -\frac{c(xU_k P_{k+1}^{(0)})}{c^{(1)}(U_k Q_k)} = \frac{(\boldsymbol{A}U_k(\boldsymbol{A})\boldsymbol{r}_{k+1}^{(0)}, \boldsymbol{y})}{(\boldsymbol{A}U_k(\boldsymbol{A})\boldsymbol{q}_k, \boldsymbol{y})} \tag{IV.44}$$

$$\lambda_k^{(0)} = \frac{(\boldsymbol{A}^k \boldsymbol{r}_k^{(0)}, \boldsymbol{y})}{(\boldsymbol{A}^{k+1}\boldsymbol{q}_k, \boldsymbol{y})} = \frac{c(V_k^{(0)} P_k^{(0)})}{c^{(1)}(V_k^{(0)} Q_k)} = \frac{(V_k^{(0)}(\boldsymbol{A})\boldsymbol{r}_k^{(0)}, \boldsymbol{y})}{(\boldsymbol{A}V_k^{(0)}(\boldsymbol{A})\boldsymbol{q}_k, \boldsymbol{y})}.$$

Preuve :

La démonstration est uniquement basée sur les conditions d'orthogonalité que vérifient les polynômes $P_k^{(0)}$ et Q_k ainsi que sur la linéarité des fonctionnelles c et $c^{(1)}$.

■

On remarque que l'expression de $\lambda_k^{(0)}$ est compatible avec celle obtenue en (IV.41) si $j = 0$.

Comme dans le cas précédent, si l'on choisit $U_k(x) = x^k$ et $V_k^{(j)}(x) = x^k$ alors la mise en œuvre est immédiate et l'algorithme Multiple Lanczos/Orthomin (noté M–Lanczos/Orthomin) est ainsi obtenu.

Algorithme M–Lanczos/Orthomin$(\boldsymbol{A}, B, X_0, \boldsymbol{x}_0^{(0)}, \boldsymbol{y}, \boldsymbol{z}, \varepsilon)$

• **Initialisations**

$\boldsymbol{q}_0 = \boldsymbol{z}$
$\boldsymbol{b}^{(0)} \leftarrow \boldsymbol{A}\boldsymbol{x}_0^{(0)} - \boldsymbol{z}$
$R_0 \leftarrow B - \boldsymbol{A}X_0$
$\boldsymbol{r}_0^{(0)} \leftarrow \boldsymbol{b}^{(0)} - \boldsymbol{A}\boldsymbol{x}_0^{(0)} = -\boldsymbol{z}$
$\boldsymbol{y}_0 = \boldsymbol{y}$

• **Itérations**

Pour $k = 0, \ldots$ jusque convergence **Faire**
$\widetilde{\boldsymbol{q}}_k = \boldsymbol{A}\boldsymbol{q}_k$
$d_k = (\widetilde{\boldsymbol{q}}_k, \boldsymbol{y}_k)$
Pour $j = 0, \ldots, s$ **Faire**
$\lambda_k^{(j)} \leftarrow (\boldsymbol{r}_k^{(j)}, \boldsymbol{y}_k)/d_k$
$\boldsymbol{r}_{k+1}^{(j)} \leftarrow \boldsymbol{r}_k^{(j)} - \lambda_k^{(j)}\widetilde{\boldsymbol{q}}_k$
$\boldsymbol{x}_{k+1}^{(j)} \leftarrow \boldsymbol{x}_k^{(j)} + \lambda_k^{(j)}\boldsymbol{q}_k$
Fin de Pour.
Si $\max_{1 \leqslant j \leqslant s} \left\| \boldsymbol{r}_{k+1}^{(j)} \right\| \leqslant \varepsilon$ **Alors** Stop.
$\boldsymbol{y}_{k+1} \leftarrow \boldsymbol{A}^*\boldsymbol{y}_k$
$\gamma_k \leftarrow (\boldsymbol{r}_{k+1}^{(0)}, \boldsymbol{y}_{k+1})/d_k$
$\boldsymbol{q}_{k+1} \leftarrow \gamma_k\boldsymbol{q}_k - \boldsymbol{r}_{k+1}^{(0)}$
Fin de Pour.

Une remarque identique à celle formulée pour l'algorithme M-Lanczos/Orthodir concernant le calcul des puissances itérées de $\boldsymbol{A}^*$ peut être énoncée ici.

C'est pourquoi, si l'on pose à nouveau, $U_k \equiv Q_k$ et $V_k^{(j)} \equiv Q_k$, alors on obtient la

Proposition 2.6
En posant $\bar{\boldsymbol{r}}_k^{(j)} = Q_k(\boldsymbol{A})\boldsymbol{r}_k^{(j)}$, $\bar{\boldsymbol{q}}_k = Q_k(\boldsymbol{A})\boldsymbol{q}_k$, $\widehat{\boldsymbol{r}}_k^{(j)} = Q_{k-1}(\boldsymbol{A})\boldsymbol{r}_k^{(j)}$, *et* $\ddot{\boldsymbol{r}}_k^{(j)} = P_k^{(0)}(\boldsymbol{A})\boldsymbol{r}_k^{(j)}$, *on a*

$$\begin{aligned}
\ddot{\boldsymbol{r}}_{k+1}^{(j)} &= \ddot{\boldsymbol{r}}_k^{(j)} + \lambda_k^{(j)}\boldsymbol{A}\bar{\boldsymbol{r}}_k^{(0)} + \lambda_k^{(0)}\boldsymbol{A}\bar{\boldsymbol{r}}_k^{(j)} - \lambda_k^{(0)}\lambda_k^{(j)}\boldsymbol{A}^2\bar{\boldsymbol{q}}_k \qquad \text{(IV.45)}\\
\widehat{\boldsymbol{r}}_{k+1}^{(j)} &= \bar{\boldsymbol{r}}_k^{(j)} - \lambda_k^{(j)}\boldsymbol{A}\bar{\boldsymbol{q}}_k\\
\bar{\boldsymbol{r}}_{k+1}^{(j)} &= \ddot{\boldsymbol{r}}_{k+1}^{(j)} + \gamma_k\widehat{\boldsymbol{r}}_{k+1}^{(j)}\\
\bar{\boldsymbol{q}}_{k+1} &= \gamma_k^2\bar{\boldsymbol{q}}_k - \ddot{\boldsymbol{r}}_{k+1}^{(0)} - 2\gamma_k\widehat{\boldsymbol{r}}_{k+1}^{(0)}
\end{aligned}$$

avec

$$\begin{aligned}
\lambda_k^{(j)} &= \frac{(\bar{\boldsymbol{r}}_k^{(j)}, \boldsymbol{y})}{(\boldsymbol{A}\bar{\boldsymbol{q}}_k, \boldsymbol{y})}\\
\gamma_k &= \frac{(\boldsymbol{A}\widehat{\boldsymbol{r}}_{k+1}^{(0)}, \boldsymbol{y})}{(\boldsymbol{A}\bar{\boldsymbol{q}}_k, \boldsymbol{y})}.
\end{aligned}$$

Preuve :

Par définition, on a $\ddot{\boldsymbol{r}}_{k+1}^{(j)} = P_{k+1}^{(0)}(\boldsymbol{A})\boldsymbol{r}_{k+1}^{(j)}$. En utilisant la relation (IV.38) et la relation (IV.42), on obtient $\ddot{\boldsymbol{r}}_{k+1}^{(j)} = \left[P_k^{(0)}(\boldsymbol{A}) + \lambda_k^{(0)}\boldsymbol{A}\widetilde{P}_k(\boldsymbol{A})\right]\left[\boldsymbol{r}_k^{(j)} - \lambda_k^{(j)}\boldsymbol{A}\boldsymbol{q}_k\right]$. Ainsi, en développant et en remarquant, comme $P_k^{(0)}(\boldsymbol{A})\boldsymbol{z} = -\boldsymbol{r}_k^{(0)}$, que $P_k^{(0)}(\boldsymbol{A})\boldsymbol{q}_k = -\widetilde{P}_k(\boldsymbol{A})\boldsymbol{r}_k^{(0)}$, on obtient le résultat annoncé.

Par définition de $\widehat{\boldsymbol{r}}_{k+1}^{(j)}$, en utilisant (IV.38) et le fait que $\boldsymbol{r}_{k+1}^{(0)} = \boldsymbol{r}_k^{(0)} - \lambda_k^{(0)}\boldsymbol{A}\boldsymbol{q}_k$, on trouve que $\widehat{\boldsymbol{r}}_{k+1}^{(j)} = \widetilde{P}_k(\boldsymbol{A})\left[\boldsymbol{r}_k^{(j)} - \lambda_k^{(j)}\boldsymbol{A}\boldsymbol{q}_k\right]$, d'où le résultat.

Par la définition de $\bar{\boldsymbol{r}}_{k+1}^{(j)}$ et en utilisant la relation de récurrence (IV.43), on trouve que $\bar{\boldsymbol{r}}_{k+1}^{(j)} = \left[P_{k+1}^{(0)}(\boldsymbol{A}) + \gamma_k\widetilde{P}_k(\boldsymbol{A})\right]\boldsymbol{r}_{k+1}^{(j)}$. En développant, l'expression de $\bar{\boldsymbol{r}}_{k+1}^{(j)}$ est immédiate.

Par définition de $\bar{\boldsymbol{q}}_{k+1}$, on a $\bar{\boldsymbol{q}}_{k+1} = \widetilde{P}_{k+1}^2(\boldsymbol{A})\boldsymbol{z}$. En utilisant la relation de récurrence (IV.43), on obtient $\bar{\boldsymbol{q}}_{k+1} = \left[P_{k+1}^{(0)}(\boldsymbol{A}) + \gamma_k\widetilde{P}_k(\boldsymbol{A})\right]^2\boldsymbol{z}$. En développant le carré et en remarquant à nouveau que $\boldsymbol{r}_{k+1}^{(0)} = -P_{k+1}^{(0)}(\boldsymbol{A})\boldsymbol{z}$, l'expression de

$\bar{\boldsymbol{q}}_{k+1}$ est obtenue.

L'expression des coefficients $\lambda_k^{(j)}$ et γ_k est trivialement obtenue en remplaçant U_k et $V_k^{(j)}$ tous deux par $\widetilde{P}_k$ dans (IV.41) et (IV.44).

■

Proposition 2.7
Comme $P_k^{(0)}(0) = 1$ *alors on peut écrire* $\ddot{\boldsymbol{r}}_k^{(j)} = \boldsymbol{b}^{(j)} - \boldsymbol{A}\ddot{\boldsymbol{x}}_k^{(j)}$ *et*

$$\boldsymbol{r}_k^{(j)} = 0 \Longrightarrow \ddot{\boldsymbol{r}}_k^{(j)} = 0 \Longrightarrow \boldsymbol{A}\ddot{\boldsymbol{x}}_k^{(j)} = \boldsymbol{b}^{(j)}.$$

De plus

$$\ddot{\boldsymbol{x}}_{k+1}^{(j)} = \ddot{\boldsymbol{x}}_k^{(j)} - \lambda_k^{(j)}\bar{\boldsymbol{r}}_k^{(0)} - \lambda_k^{(0)}\bar{\boldsymbol{r}}_k^{(j)} + \lambda_k^{(0)}\lambda_k^{(j)}\boldsymbol{A}\bar{\boldsymbol{q}}_k.$$

Preuve :

Par simple définition de $\ddot{\boldsymbol{r}}_k^{(j)}$, si $\boldsymbol{r}_k^{(j)} = 0$, il est alors clair que $\ddot{\boldsymbol{r}}_k^{(j)} = 0$. D'autre part, on remarque que , comme $P_k^{(0)}(0) = 1$, $\ddot{\boldsymbol{r}}_k^{(j)} = P_k^{(0)}(\boldsymbol{A})\boldsymbol{r}_k^{(j)} = (\boldsymbol{I} + \boldsymbol{A}\Phi_{k-1}^{(0)}(\boldsymbol{A}))\boldsymbol{r}_k^{(j)} = (\boldsymbol{I} + \boldsymbol{A}\Phi_{k-1}^{(0)}(\boldsymbol{A}))(\boldsymbol{b}^{(j)} - \boldsymbol{A}\boldsymbol{x}_k^{(j)}) = \boldsymbol{b}^{(j)} - \boldsymbol{A}\ddot{\boldsymbol{x}}_k^{(j)}$. Alors on peut obtenir par récurrence $\ddot{\boldsymbol{x}}_k^{(j)}$ en formant, à partir de (IV.45), $\ddot{\boldsymbol{r}}_k^{(j)} - \ddot{\boldsymbol{r}}_{k+1}^{(j)}$ et en multipliant par $\boldsymbol{A}^{-1}$.

■

Ainsi, il n'est pas nécessaire de calculer les vecteurs $\boldsymbol{r}_k^{(j)}$ ainsi que les vecteurs $\boldsymbol{x}_k^{(j)}$, ce qui est intéressant du point de vue coût opératoire et encombrement mémoire. De plus, cette fois, les vecteurs $\ddot{\boldsymbol{x}}_k^{(j)}$ ne sont pas les mêmes que ceux évalués avec l'algorithme M–Lanczos/Orthomin.

Un algorithme sans utilisation de la transposée s'en suit. Il s'agit de TFM–Lanczos/Orthomin (pour Transpose Free Multiple Lanczos/Orthomin).

Algorithme TFM–Lanczos/Orthomin$(\boldsymbol{A}, B, X_0, \boldsymbol{x}_0^{(0)}, \boldsymbol{y}, \boldsymbol{z}, \varepsilon)$

- **Initialisations**

$\boldsymbol{b}^{(0)} \leftarrow \boldsymbol{A}\boldsymbol{x}_0^{(0)} - \boldsymbol{z}$
$\bar{\boldsymbol{q}}_0 = \boldsymbol{z}$
$\ddot{X}_0 = X_0$
$\ddot{\boldsymbol{x}}_0^{(0)} = \boldsymbol{x}_0^{(0)}$

$\bar{R}_0 \leftarrow B - \boldsymbol{A}X_0$
$\ddot{R}_0 = \bar{R}_0$
$\ddot{\boldsymbol{r}}_0^{(0)} = -\boldsymbol{z}$
$\bar{\boldsymbol{r}}_0^{(0)} = \ddot{\boldsymbol{r}}_0^{(0)}$

- **Itérations**

Pour $k = 0, \ldots$ jusque convergence **Faire**
$\widetilde{\boldsymbol{q}}_k \leftarrow \boldsymbol{A}\bar{\boldsymbol{q}}_k$
$d_k \leftarrow (\widetilde{\boldsymbol{q}}_k, \boldsymbol{y})$
Pour $j = 0, \ldots, s$ **Faire**
$\widetilde{\boldsymbol{r}}_k \leftarrow \boldsymbol{A}\bar{\boldsymbol{r}}_k^{(j)}$
$\lambda_k^{(j)} \leftarrow (\bar{\boldsymbol{r}}_k^{(j)}, \boldsymbol{y})/d_k$
$\widehat{\boldsymbol{r}}_{k+1} \leftarrow \bar{\boldsymbol{r}}_k^{(j)} - \lambda_k \widetilde{\boldsymbol{q}}_k$
Si $j = 0$ **Alors**
$\widetilde{\boldsymbol{r}}_k^{(0)} = \widetilde{\boldsymbol{r}}_k$
$\widehat{\boldsymbol{r}}_{k+1}^{(0)} = \widehat{\boldsymbol{r}}_{k+1}$
$\gamma_k \leftarrow (\boldsymbol{A}\widehat{\boldsymbol{r}}_{k+1}^{(0)}, \boldsymbol{y})/d_k$
$\lambda_k^{(0)} = \lambda_k$
Fin de Si.
$\ddot{\boldsymbol{r}}_{k+1}^{(j)} \leftarrow \ddot{\boldsymbol{r}}_k^{(j)} + \lambda_k \widetilde{\boldsymbol{r}}_k^{(0)} + \lambda_k^{(0)} \widetilde{\boldsymbol{r}}_k - \lambda_k^{(0)} \lambda_k \boldsymbol{A}\widetilde{\boldsymbol{q}}_k$
$\bar{\boldsymbol{r}}_{k+1}^{(j)} \leftarrow \ddot{\boldsymbol{r}}_{k+1}^{(j)} + \gamma_k \widehat{\boldsymbol{r}}_{k+1}$
$\ddot{\boldsymbol{x}}_{k+1}^{(j)} \leftarrow \ddot{\boldsymbol{x}}_k^{(j)} - \lambda_k \bar{\boldsymbol{r}}_k^{(0)} - \lambda_k^{(0)} \bar{\boldsymbol{r}}_k^{(j)} + \lambda_k^{(0)} \lambda_k \widetilde{\boldsymbol{q}}_k$
Fin de Pour.
Si $\max_{1 \leqslant j \leqslant s} \left\| \ddot{\boldsymbol{r}}_{k+1}^{(j)} \right\| \leqslant \varepsilon$ **Alors** Stop.
$\bar{\boldsymbol{q}}_{k+1} \leftarrow \gamma_k^2 \bar{\boldsymbol{q}}_k - \ddot{\boldsymbol{r}}_{k+1}^{(0)} - 2\gamma_k \widehat{\boldsymbol{r}}_{k+1}^{(0)}$
Fin de Pour.

Analogie avec le BiCGStab Supposons maintenant que, pour tout j, on ait $V_k^{(j)} = V_k$ défini par récurrence par

$$V_{k+1}(x) = (1 + \nu_k x) V_k(x)$$

où ν_k minimise

$$\sum_{j=1}^{s} \left\| \bar{\boldsymbol{r}}_{k+1}^{(j)} \right\|^2$$

et où $\bar{\boldsymbol{r}}_k^{(j)}$ est défini par

$$\bar{\boldsymbol{r}}_k^{(j)} = V_k(\boldsymbol{A}) \boldsymbol{r}_k^{(j)}.$$

Proposition 2.8

Si $\boldsymbol{q}_k = Q_k(\boldsymbol{A})\boldsymbol{z}$, *en posant* $\bar{\boldsymbol{r}}_k^{(j)} = V_k(\boldsymbol{A})\boldsymbol{r}_k^{(j)}$, $\bar{\boldsymbol{q}}_k = V_k(\boldsymbol{A})\boldsymbol{q}_k$ *et* $\bar{\boldsymbol{s}}_k^{(j)} = \bar{\boldsymbol{r}}_k^{(j)} - \lambda_k^{(j)}\boldsymbol{A}\bar{\boldsymbol{q}}_k$, *on obtient*

$$\begin{aligned} \bar{\boldsymbol{q}}_{k+1} &= \gamma_k(\boldsymbol{I} + \nu_k\boldsymbol{A})\bar{\boldsymbol{q}}_k - \bar{\boldsymbol{r}}_{k+1}^{(0)} \\ \bar{\boldsymbol{r}}_{k+1}^{(j)} &= (\boldsymbol{I} + \nu_k\boldsymbol{A})\bar{\boldsymbol{s}}_k^{(j)}. \end{aligned} \tag{IV.46}$$

avec

$$\begin{aligned} \lambda_k^{(j)} &= \frac{(\bar{\boldsymbol{r}}_k^{(j)}, \boldsymbol{y})}{(\boldsymbol{A}\bar{\boldsymbol{q}}_k, \boldsymbol{y})} \\ \nu_k &= -\frac{\sum_{j=1}^{s}(\bar{\boldsymbol{s}}_k^{(j)}, \boldsymbol{A}\bar{\boldsymbol{s}}_k^{(j)})}{\sum_{j=1}^{s}\left\|\boldsymbol{A}\bar{\boldsymbol{s}}_k^{(j)}\right\|^2} \\ \gamma_k &= \frac{1}{\nu_k}\frac{(\bar{\boldsymbol{r}}_{k+1}^{(0)}, \boldsymbol{y})}{(\boldsymbol{A}\bar{\boldsymbol{q}}_k, \boldsymbol{y})}. \end{aligned}$$

Preuve :

Par la relation de récurrence introduite en (IV.43), on a $\boldsymbol{q}_{k+1} = \gamma_k\boldsymbol{q}_k - \boldsymbol{r}_{k+1}^{(0)}$ et ainsi, $\bar{\boldsymbol{q}}_{k+1} = \gamma_k(\boldsymbol{I} + \nu_k\boldsymbol{A})\bar{\boldsymbol{q}}_k - \bar{\boldsymbol{r}}_{k+1}^{(0)}$ si l'on considère la relation que vérifie le polynôme V_k.

La définition de $\bar{\boldsymbol{s}}_k^{(j)}$, celle de V_k et (IV.38) donnent l'expression de $\bar{\boldsymbol{r}}_{k+1}^{(j)}$.

L'expression de $\lambda_k^{(j)}$ vient de (IV.41) et les conditions de minimisation imposées donnent ν_k.

Pour γ_k, on utilise (IV.44). Et l'on remarque que, par les conditions d'orthogonalité que vérifie $\boldsymbol{r}_k^{(0)}$, on a $\nu_k(\boldsymbol{A}V_k(\boldsymbol{A})\boldsymbol{r}_{k+1}^{(0)}, \boldsymbol{y}) = (V_k(\boldsymbol{A})\boldsymbol{r}_{k+1}^{(0)}, \boldsymbol{y}) + \nu_k(\boldsymbol{A}V_k(\boldsymbol{A})\boldsymbol{r}_{k+1}^{(0)}, \boldsymbol{y}) = (V_{k+1}(\boldsymbol{A})\boldsymbol{r}_{k+1}^{(0)}, \boldsymbol{y})$. Le dénominateur ne change pas et l'on obtient l'expression voulue de γ_k.

■

Comme pour TFM–Lanczos/Orthomin, on la

Proposition 2.9

Comme $V_k(0) = 1$, *alors on peut écrire* $\bar{\boldsymbol{r}}_k = \boldsymbol{b}^{(j)} - \boldsymbol{A}\bar{\boldsymbol{x}}_k^{(j)}$ *et*

$$\boldsymbol{r}_k^{(j)} = 0 \Longrightarrow \bar{\boldsymbol{r}}_k^{(j)} = 0 \Longrightarrow \boldsymbol{A}\bar{\boldsymbol{x}}_k^{(j)} = \boldsymbol{b}^{(j)}.$$

De plus,

$$\bar{\boldsymbol{x}}_{k+1}^{(j)} = \bar{\boldsymbol{x}}_k^{(j)} + \lambda_k^{(j)} \bar{\boldsymbol{q}}_k - \nu_k \bar{\boldsymbol{s}}_k^{(j)}.$$

Preuve :

La démonstration est strictement identique à celle de la Proposition 2.7 mais on utilise cette fois (IV.46) pour l'expression de $\bar{\boldsymbol{x}}_k^{(j)}$.

■

Ainsi l'algorithme TFM–BiCGStab/Orthomin est obtenu.

Algorithme TFM–BiCGStab/Orthomin$(\boldsymbol{A}, B, X_0, \boldsymbol{x}_0^{(0)}, \boldsymbol{y}, \boldsymbol{z}, \varepsilon)$

- **Initialisations**

  $\boldsymbol{b}^{(0)} \leftarrow \boldsymbol{A}\boldsymbol{x}_0^{(0)} - \boldsymbol{z}$
  $\bar{\boldsymbol{q}}_0 = \boldsymbol{z}$
  $\bar{X}_0 = X_0$
  $\bar{\boldsymbol{x}}_0^{(0)} = \boldsymbol{x}_0^{(0)}$
  $\bar{R}_0 \leftarrow B - \boldsymbol{A}X_0$
  $\bar{\boldsymbol{r}}_0 \leftarrow \boldsymbol{b} - \boldsymbol{A}\bar{\boldsymbol{x}}_0^{(0)} = -\boldsymbol{z}$

- **Itérations**

  Pour $k = 0, \ldots$ jusque convergence **Faire**
    $\widetilde{\boldsymbol{q}}_k \leftarrow \boldsymbol{A}\bar{\boldsymbol{q}}_k$
    $d_k \leftarrow (\widetilde{\boldsymbol{q}}_k, \boldsymbol{y})$
    Pour $j = 0, \ldots, s$ **Faire**
      $\lambda_k^{(j)} \leftarrow (\bar{\boldsymbol{r}}_k^{(j)}, \boldsymbol{y})/d_k$
      $\bar{\boldsymbol{s}}_k^{(j)} \leftarrow \bar{\boldsymbol{r}}_k^{(j)} - \lambda_k^{(j)} \widetilde{\boldsymbol{q}}_k$
      $\widetilde{\boldsymbol{s}}_k^{(j)} \leftarrow \boldsymbol{A}\bar{\boldsymbol{s}}_k^{(j)}$
    Fin de Pour.
    $\nu_k \leftarrow -\sum_{j=1}^{s} (\bar{\boldsymbol{s}}_k^{(j)}, \widetilde{\boldsymbol{s}}_k^{(j)}) / \sum_{j=1}^{s} \left\| \widetilde{\boldsymbol{s}}_k^{(j)} \right\|^2$
    Pour $j = 0, \ldots, s$ **Faire**
      $\bar{\boldsymbol{r}}_{k+1}^{(j)} \leftarrow \bar{\boldsymbol{s}}_k^{(j)} + \nu_k \widetilde{\boldsymbol{s}}_k^{(j)}$
      $\bar{\boldsymbol{x}}_{k+1}^{(j)} \leftarrow \bar{\boldsymbol{x}}_k^{(j)} + \lambda_k^{(j)} \bar{\boldsymbol{q}}_k - \nu_k \bar{\boldsymbol{s}}_k^{(j)}$
    Fin de Pour.
    Si $\max\limits_{1 \leqslant j \leqslant s} \left\| \bar{\boldsymbol{r}}_{k+1}^{(j)} \right\| \leqslant \varepsilon$ **Alors** Stop.
    $\gamma_k = \frac{1}{\nu_k} (\bar{\boldsymbol{r}}_{k+1}^{(0)}, \boldsymbol{y})/d_k$
    $\bar{\boldsymbol{q}}_{k+1} \leftarrow \gamma_k (\bar{\boldsymbol{q}}_k + \nu_k \widetilde{\boldsymbol{q}}_k) - \bar{\boldsymbol{r}}_{k+1}^{(0)}$.

Fin de Pour.

Remarque 2.5

Une analogie avec Lanczos/Orthores ne peut pas être obtenue car les polynômes $P_k^{(j)}$ ne sont pas en général des polynômes orthogonaux. Or, cette propriété était utilisée pour les formules de récurrence attachées à Lanczos/Orthores.

Le tableau 21 montre le nombre de produits scalaires, le nombre de produits matrice-vecteur par itération (produits Mv) ainsi que l'encombrement mémoire (nombre de vecteurs de dimension n) nécessaire à la mise en œuvre de chaque méthode.

	Produits scalaires	Produits Mv	Mémoire
M–Lanczos/Orthodir	$s+2$	2	$2s+2$
M–Lanczos/Orthomin	$s+3$	2	$2s+4$
TFM–Lanczos/Orthodir	$s+4$	$2s+4$	$4s+7$
TFM–Lanczos/Orthomin	$s+3$	$s+4$	$3s+8$
TFM–BiCGStab/Orthomin	$3s+3$	$s+2$	$4s+6$

TAB. 21: *Coût opératoire par itération (produits scalaires et produits matrice-vecteur) et emplacement mémoire requis selon l'algorithme considéré.*

Ce tableau montre que le nombre de produits matrice-vecteur ne dépend pas de s pour les deux premiers algorithmes alors qu'il n'en est pas de même pour les algorithmes sans utilisation de la transposée. Il faut également remarquer que les algorithmes sans utilisation de la transposée nécessitent plus de mémoire et qu'ils peuvent être rapprochés des algorithmes trouvés dans [16] et [28].

La méthode proposée ne requiert, pour les mises en œuvre de base (M–Lanczos/Orthodir et M–Lanczos/Orthomin) que deux produits matrice-vecteur par itération (indépendamment du nombre de seconds membres considérés s).

Malheureusement, plus n est grand, plus les résultats numériques correspondant aux algorithmes de base sont mauvais (voir la Section 4). Ces deux algorithmes ne semblent alors avoir qu'un intérêt théorique, à moins d'en améliorer la convergence.

De plus, l'algorithme TFM–Lanczos/Orthodir semble prohibitif vu le nombre de produits matrice-vecteur qu'il requiert par rapport aux autres. Pour les deux derniers algorithmes (TFM–Lanczos/Orthomin et TFM-BiCGStab/Orthomin), l'un présente un avantage au niveau du nombre de produits matrice-vecteur tandis que l'autre demande moins de produits scalaires.

Remarque 2.6

Une forme de Conjugate Gradient Square (CGS) [68] ne peut pas être mise en œuvre ici car nous nous sommes contentés de polynômes V_k indépendants de j, bien que TFM–Lanczos/Orthomin puisse y faire penser (puisque l'on utilise $P_k^{(0)^2}$). Pour le CGS, il faudrait considérer $P_k^{(j)^2}$ pour $1 \leqslant j \leqslant s$.

Remarque 2.7

Évidemment, dans tous les algorithmes de cette partie, au lieu que le test d'arrêt ne porte sur le maximum des résidus obtenus, il peut porter, par exemple, sur la moyenne de ceux-ci ou sur la norme de Frobenius de ces derniers. Il peut également être défini par rapport aux seconds membres (norme des résidus sur celle des seconds membres) ...

3 Exemples numériques

Les exemples numériques seront de trois types. Dans une **première sous-section** nous étudierons la répercussion du nombre de seconds membres sur les algorithmes de la Section 2. Des graphes permettant une telle comparaison seront alors donnés et commentés.

Tandis que dans la **deuxième sous-section** les trois algorithmes *Block Bi-CG*, TFM–Lanczos/Orthomin et TFM–BiCGStab/Orthomin seront comparés, dans un certain sens que nous préciserons. Cette comparaison sera menée afin de mieux apprécier la convergence des algorithmes obtenus dans la Section 2 par rapport, par exemple, à une méthode de résolution par bloc.

Enfin, dans la **troisième sous-section**, nous utiliserons des matrices creuses de grandes dimensions afin de voir l'incidence de la dimension des matrices sur les algorithmes de la Section 2.

3.1 Incidence du nombre de seconds membres sur la convergence

Avant de considérer les exemples numériques, remarquons premièrement que les méthodes M–Lanczos/Orthomin et M–Lanczos/Orthodir ont paru n'être efficace que sur des matrices de très petite dimension (inférieure à 20!), même si le coût opératoire est de loin le plus faible par rapport au TFM–BiCGStab/Orthomin et aux deux autres algorithmes sans utilisation de la transposée. Cela est probablement dû au calcul des puissances itérées de $\boldsymbol{A}^*$.

Deuxièmement, le TFM–Lanczos/Orthodir semble n'être efficace que sur des matrices de dimension inférieure à 100. Cela est peut-être dû au fait qu'un grand nombre de produits matrice-vecteur est considéré.

C'est pourquoi nous n'étudierons que des exemples numériques utilisant le TFM–BiCGStab/Orthomin et le TFM–Lanczos/Orthomin. Seuls ces deux algorithmes ont en effet donné des résultats encourageants lors de leurs mises en œuvre.

Tous les algorithmes ont été écrits et programmés en *Matlab* 4.2*c*.1 en double précision. Toutes les matrices utilisées sont de dimension $n = 500$ pour cette sous-section. Les seconds membres ont été choisis au hasard, en utilisant la fonction *rand* de *Matlab*. Les vecteurs $\boldsymbol{x}_0^{(j)}$ sont toujours nuls. Le test d'arrêt utilisé est $\frac{1}{s}\sum_{i=1}^{s}\left\|\boldsymbol{r}_k^{(i)}\right\|^2 < 1e^{-16}$, afin de voir le comportement de la convergence (à moins

que l'itération ne corresponde à la dimension de la matrice, c'est-à-dire 500, auquel cas l'algorithme a été arrêté).

Nous avons utilisé trois matrices symétriques et trois matrices non symétriques afin de voir la rapidité de convergence dans les deux cas. Ces deux types de matrices sont donc regroupées selon ce critère. Pour chaque matrice, le conditionnement a été calculé à l'aide de la fonction *cond* de *Matlab*, qui est définie comme le rapport de la plus grande valeur singulière sur la plus petite (le conditionnement en norme 2).

Ensuite, nous avons considéré, si s désigne le nombre de seconds membres, $s = 1, 10, 20, 30, 40$ et 50 pour voir le comportement du TFM–BiCGStab/Orthomin en fonction du nombre de seconds membres considérés (puisque le coefficient ν_k dépend de tous les résidus et donc du nombre de seconds membres). Il ne paraît pas nécessaire d'effectuer une telle comparaison pour le TFM–Lanczos/Orthomin car chaque résidu est considéré indépendamment des autres, contrairement à TFM–BiCGStab/Orthomin où le coefficient ν_k dépend de tous les résidus. Tous les résultats sont présentés sous forme de tableau pour la méthode M-BiCGStab/Orthomin dans la mesure où l'on peut ainsi apprécier l'incidence du nombre de seconds membres sur la mise en œuvre de la méthode.

De plus, pour chaque matrice, un graphique contient les résultats pour $s = 1$ et $s = 50$ pour le TFM–BiCGStab/Orthomin et pour $s = 50$ pour le TFM–Lanczos/Orthomin. Ceci nous permettra de comparer le comportement du TFM–BiCGStab/Orthomin pas à pas lorsque l'on augmente le nombre de seconds membres. Le TFM–BiCGStab/Orthomin et le TFM–Lanczos/Orthomin pourront être comparés également. Les graphes indiquent, avec une échelle logarithmique pour les ordonnées, la moyenne de la norme euclidienne des résidus obtenue à chaque itération. Dans les graphiques, on peut lire en abscisse l'itération pour laquelle la moyenne des normes des résidus est calculée.

3.1.1 Les matrices symétriques

Nous allons tout d'abord étudier la mise en œuvre du TFM–BiCGStab/Orthomin et de TFM–Lanczos/Orthomin à l'aide de matrices symétriques.

La première matrice considérée est la matrice

$$M_1 = \begin{pmatrix} 20 & -1 & & \\ -1 & 20 & \ddots & \\ & \ddots & \ddots & -1 \\ & & -1 & 20 \end{pmatrix}.$$

Elle est de dimension 500 et son conditionnement est 1.22. Les données du tableau 22 donnent les résultats obtenus (itération où le critère d'arrêt est atteint) pour le TFM–BiCGStab/Orthomin.

s	1	10	20	30	40	50
Itérations	16	16	16	17	18	16

TAB. 22: *Nombre d'itérations nécessaires à la satisfaction du critère d'arrêt pour la matrice M_1 pour la méthode du TFM-BiCGStab/Orthomin avec successivement* 1, 10, 20, 30, 40 *puis* 50 *seconds membres.*

La convergence pour $s = 1$ et $s = 50$ a le comportement décrit par la figure 1.

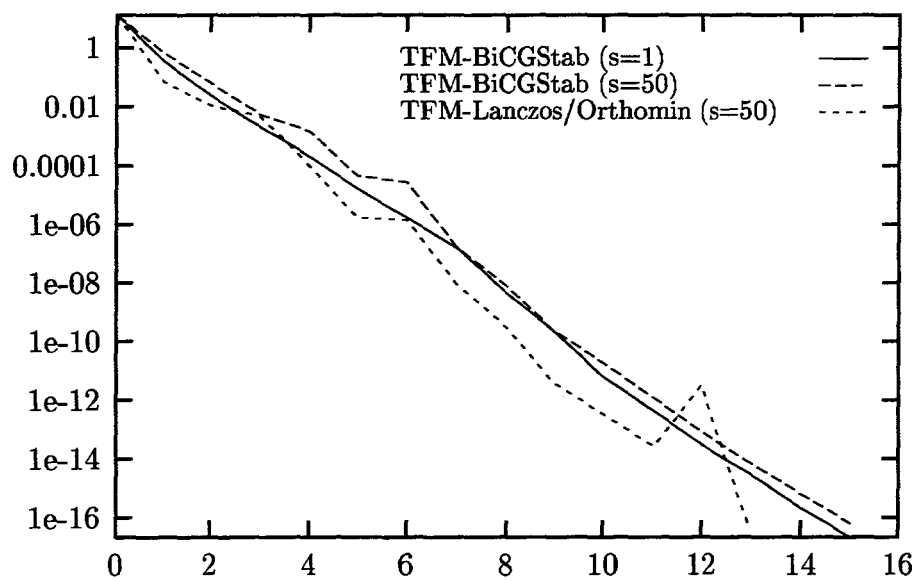

FIG. 1: *Résultats obtenus pour la méthode du TFM–BiCGStab avec* 1 *puis* 50 *seconds membres et pour TFM–Lanczos/Orthomin (*50 *seconds membres uniquement) pour la matrice M_1.*

La convergence est rapide quelle que soit la méthode. Cela est sûrement dû à deux facteurs. Premièrement, la matrice considérée est symétrique. Deuxièmement, son conditionnement est très bon. Ce qui semble intéressant est que la courbe a une allure générale semblable, que $s = 1$ ou que $s = 50$ pour le TFM–BiCGStab/Orthomin. TFM–Lanczos/Orthomin nous donne de bons résultats également.

La matrice suivante est

$$M_2 = \begin{pmatrix} B & -I & & \\ -I & B & \ddots & \\ & \ddots & \ddots & -I \\ & & -I & B \end{pmatrix}$$

où la matrice I est la matrice identité de dimension 20 et où $B = 4I$ est aussi de dimension 20. Le conditionnement de la matrice M_2, de dimension 500, est 2.97. Le tableau 23 consigne les données recueillies.

s	1	10	20	30	40	50
Itérations	40	45	41	41	49	43

TAB. 23: *Nombre d'itérations nécessaires à la satisfaction du critère d'arrêt pour la matrice M_2 pour la méthode du TFM-BiCGStab/Orthomin avec successivement* 1, 10, 20, 30, 40 *puis* 50 *seconds membres.*

L'allure de la convergence est représentée dans les graphes de la figure 2.

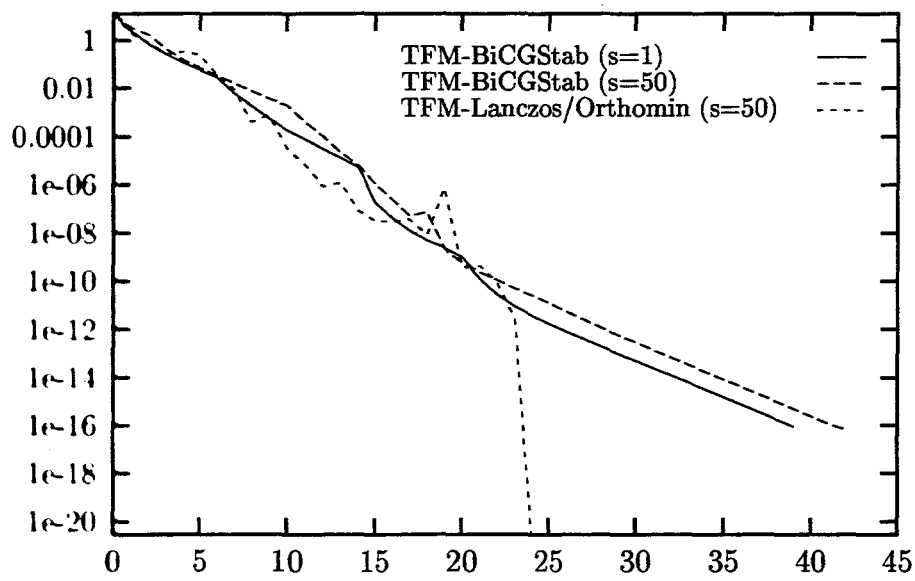

FIG. 2: *Résultats obtenus pour la méthode du TFM-BiCGStab avec* 1 *puis* 50 *seconds membres et pour TFM-Lanczos/Orthomin (*50 *seconds membres uniquement) pour la matrice M_2.*

Dans cet exemple, nous voyons que le nombre de seconds membres n'a qu'une influence minime sur le nombre d'itérations à effectuer pour arriver à la solution

correcte pour le TFM–BiCGStab/Orthomin. Le plus petit nombre d'itérations nécessaire est 40 pour $s = 1$ alors que le plus important est 49 pour $s = 40$. Les deux courbes pour le TFM–BiCGStab/Orthomin sont encore très proches. Le TFM–Lanczos/Orthomin nous donne ici un meilleur résultat.

La dernière matrice symétrique considérée est la matrice diagonale $\boldsymbol{M}_3 = \text{diag}(1, 2, \ldots, 500)$ utilisée dans [26]. Le conditionnement de cette matrice, de dimension 500, est naturellement 500. Voyons les résultats obtenus.

Ils figurent dans le tableau 24 pour $\boldsymbol{M}_3$ avec le TFM–BiCGStab/Orthomin.

s	1	10	20	30	40	50
Itérations	189	192	192	198	198	205

TAB. 24: *Nombre d'itérations nécessaires à la satisfaction du critère d'arrêt pour la matrice $\boldsymbol{M}_3$ pour la méthode du TFM-BiCGStab/Orthomin avec successivement* 1, 10, 20, 30, 40 *puis* 50 *seconds membres.*

Les graphes, pour $\boldsymbol{M}_3$, avec $s = 1$ puis $s = 50$ seconds membres sont représentés dans la figure 3.

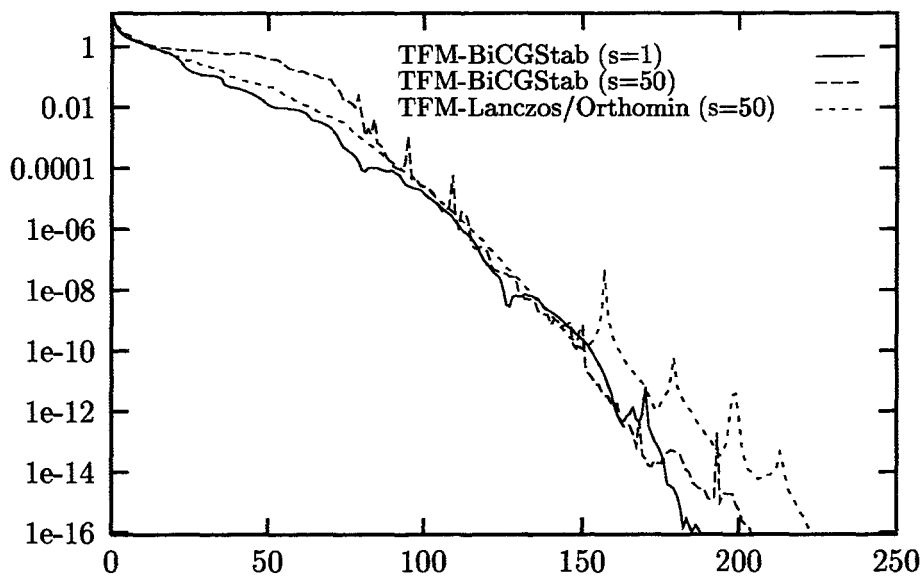

FIG. 3: *Résultats obtenus pour la méthode du TFM–BiCGStab avec* 1 *puis* 50 *seconds membres et pour TFM–Lanczos/Orthomin (*50 *seconds membres uniquement) pour la matrice $\boldsymbol{M}_3$.*

La convergence est correcte, en dépit d'un conditionnement non optimal (mais pas trop important non plus). Les deux graphes sont encore très voisins pour

le TFM–BiCGStab/Orthomin. Le nombre d'itérations minimum est 189 pour $s = 1$ tandis que le nombre d'itérations maximum est 205 pour $s = 50$. Ainsi, le nombre de seconds membres ne semble à nouveau pas être un facteur décisif de convergence ici. Le TFM–BiCGStab/Orthomin donne une convergence plus rapide que le TFM–Lanczos/Orthomin pour $s = 50$ mais les deux algorithmes ont le même comportement.

3.1.2 Matrices non symétriques

Comme le TFM–BiCGStab/Orthomin et le TFM–Lanczos/Orthomin semblent donner de bons résultats pour les matrices symétriques, on peut se demander ce qu'il va en être pour les matrices non symétriques (on sait déjà qu'en théorie, la convergence est assurée dans les deux cas).

Comme dans le cas des matrices symétriques, trois matrices seront ici étudiées. La première matrice non symétrique que nous considérons est

$$\boldsymbol{M}_4 = \begin{pmatrix} \boldsymbol{B} & -\boldsymbol{I} & & \\ -\boldsymbol{I} & \boldsymbol{B} & \ddots & \\ & \ddots & \ddots & -\boldsymbol{I} \\ & & -\boldsymbol{I} & \boldsymbol{B} \end{pmatrix}$$

avec

$$\boldsymbol{B} = \begin{pmatrix} 4 & 0 & & \\ 50 & 4 & \ddots & \\ & \ddots & \ddots & 0 \\ & & 50 & 4 \end{pmatrix}.$$

La dimension des matrices $\boldsymbol{B}$ considérées est 10. Le conditionnement de la matrice $\boldsymbol{M}_4$, de dimension 500, est 1.04×10^{14}.

On a obtenu les données du tableau 25 pour $\boldsymbol{M}_4$ avec le TFM–BiCGStab/Orthomin.

s	1	10	20	30	40	50
Itérations	138	120	158	149	153	137

TAB. 25: *Nombre d'itérations nécessaires à la satisfaction du critère d'arrêt pour la matrice $\boldsymbol{M}_4$ pour la méthode du TFM-BiCGStab/Orthomin avec successivement* 1, 10, 20, 30, 40 *puis* 50 *seconds membres.*

Le comportement de la convergence est représenté dans la figure 4 de la page suivante.

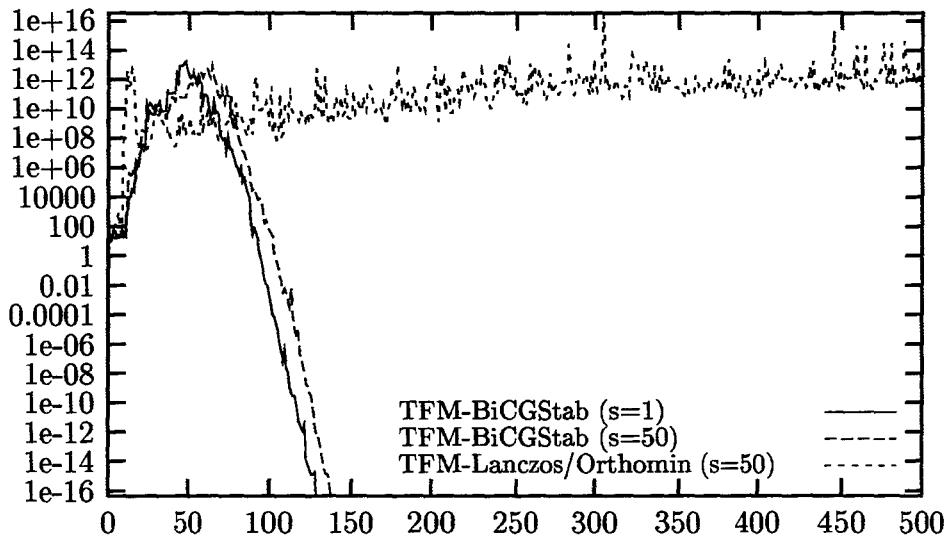

FIG. 4: *Résultats obtenus pour la méthode du TFM–BiCGStab avec* 1 *puis* 50 *seconds membres et pour TFM–Lanczos/Orthomin (*50 *seconds membres uniquement) pour la matrice* M_4.

Même si la matrice a un conditionnement assez mauvais et qu'elle est non symétrique, on peut constater que l'algorithme TFM–BiCGStab/Orthomin nous donne une convergence acceptable quel que soit le nombre de seconds membres (avec un minimum de 120 itérations pour 10 seconds membres et un maximum de 158 itérations pour 20 seconds membres). D'après le graphe, on peut également voir que le comportement de chaque courbe est très proche pour cette méthode que l'on considère 1 ou 50 seconds membres.

Par contre, pour TFM–Lanczos/Orthomin, il en va autrement dans la mesure où l'algorithme ne converge pas et que la norme des résidus oscille entre 10^8 et 10^{14}. Le conditionnement élevé de cette matrice est peut-être déterminant pour cet algorithme.

La matrice suivante utilisée est

$$M_5 = \begin{pmatrix} 2 & 1 & & & & \\ 0 & 2 & 1 & & & \\ 1 & 0 & 2 & 1 & & \\ & \ddots & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & \ddots & 1 \\ & & & 1 & 0 & 2 \end{pmatrix}$$

considérée dans [16].

Le conditionnement de cette matrice, de dimension 500, est 2.91 et les résultats obtenus sont, pour $\boldsymbol{M}_5$ et l'algorithme TFM–BiCGStab/Orthomin, représentés dans le tableau 26.

s	1	10	20	30	40	50
Itérations	274	285	284	279	286	294

TAB. 26: *Nombre d'itérations nécessaires à la satisfaction du critère d'arrêt pour la matrice $\boldsymbol{M}_5$ pour la méthode du TFM-BiCGStab/Orthomin avec successivement* 1, 10, 20, 30, 40 *puis* 50 *seconds membres.*

Le comportement de la convergence pour $\boldsymbol{M}_5$ est décrit par la figure 5.

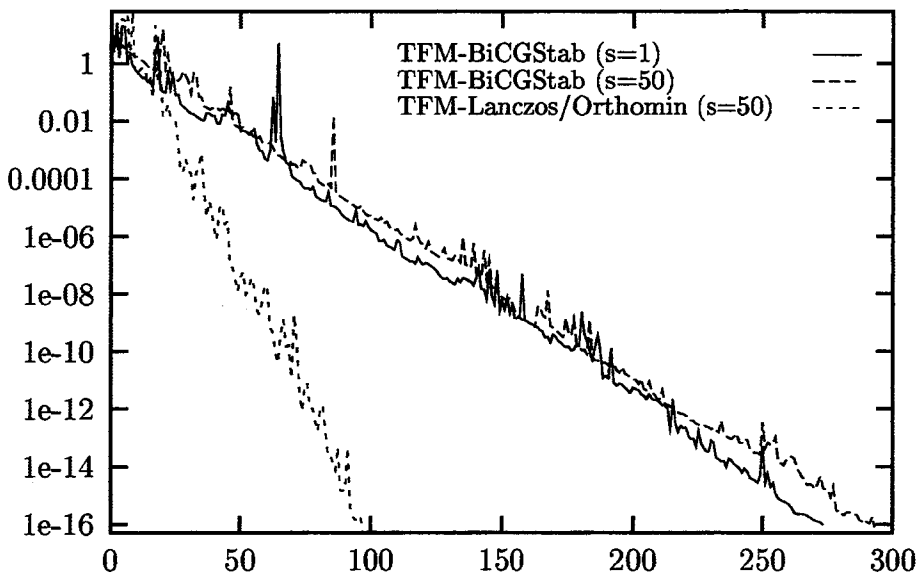

FIG. 5: *Résultats obtenus pour la méthode du TFM–BiCGStab avec* 1 *puis* 50 *seconds membres et pour TFM–Lanczos/Orthomin (*50 *seconds membres uniquement) pour la matrice $\boldsymbol{M}_5$.*

Même si le conditionnement de la matrice est petit, on a une convergence linéaire mais pas très rapide de l'algorithme TFM–BiCGStab/Orthomin. Cela doit en partie être dû au fait que la matrice est non symétrique. Toutefois, pour $s = 50$, la convergence du TFM–Lanczos/Orthomin est beaucoup plus rapide. Le nombre d'itérations nécessaire pour chaque second membre s pour le TFM–BiCGStab/Orthomin est très proche (de 274 si $s = 1$ à 294 si $s = 50$), comme on peut le constater dans le tableau.

La dernière matrice non symétrique considérée est la matrice *redheff* de la *Test Matrix Toolbox* de Higham [45] considérée également dans [16]. Si l'on écrit

cette matrice $\boldsymbol{M}_6 = (m_{i,j})_{\substack{1\leqslant i\leqslant 500\\ 1\leqslant j\leqslant 500}}$, alors les coefficients satisfont

$$m_{i,j} = 1 \text{ si } j = 1$$
$$m_{i,j} = 1 \text{ si } i \text{ divise } j$$
$$m_{i,j} = 0 \text{ sinon.}$$

Le conditionnement de cette matrice, de dimension 500, est 2.42×10^3.

On a obtenu, pour $\boldsymbol{M}_6$, les résultats du tableau 27 avec l'algorithme TFM–BiCGStab/Orthomin.

s	1	10	20	30	40	50
Itérations	50	45	47	46	48	46

TAB. 27: *Nombre d'itérations nécessaires à la satisfaction du critère d'arrêt pour la matrice $\boldsymbol{M}_6$ pour la méthode du TFM-BiCGStab/Orthomin avec successivement* 1, 10, 20, 30, 40 *puis* 50 *seconds membres.*

Et, pour $s = 1$ et $s = 50$, les graphes de la figure 6 ont été obtenus.

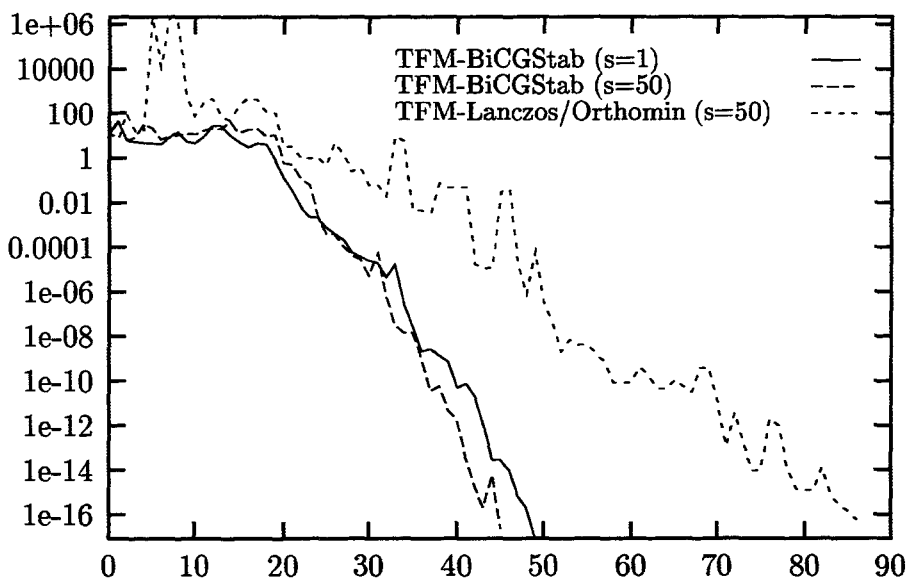

FIG. 6: *Résultats obtenus pour la méthode du TFM–BiCGStab avec* 1 *puis* 50 *seconds membres et pour TFM–Lanczos/Orthomin (*50 *seconds membres uniquement) pour la matrice $\boldsymbol{M}_6$.*

Encore une fois, le TFM–BiCGStab/Orthomin se comporte correctement pour tout s, avec un nombre d'itérations minimum de 45 pour $s = 10$ et maximum de 50 pour $s = 1$. Dans cet exemple, on peut voir que les deux courbes ont, une

nouvelle fois, une allure très semblable. Le TFM–Lanczos/Orthomin a atteint le critère d'arrêt en plus d'itérations que le TFM–BiCGStab/Orthomin.

3.2 Comparaison avec le Block Bi-CG

Dans cette sous-section, nous allons comparer les deux méthodes préalablement étudiées avec celle du *Block Bi-CG*, qui utilise la transposée de la matrice $\boldsymbol{A}$ mais qui est une référence en matière de résolution de systèmes linéaires à second membre multiple.

Même si les trois méthodes ont pour propriété commune l'obtention d'un résidu nul en un nombre fini d'itérations, la comparaison exclusive de cette donnée ne paraît pas judicieuse. En effet, pour le *Block Bi-CG* le résultat exact doit être obtenu en un maximum de $\lfloor n/s \rfloor$ itérations alors dans le cas des deux autres algorithmes il en est de même en n itérations. Ainsi, présenter les résultats obtenus sous forme de graphe n'amènerait rien de bien intéressant (si ce n'est, théoriquement, une "convergence graphique" plus rapide pour le *Block Bi-CG* mais les calculs nécessaires à cette convergence seraient plus importants pour chaque itération). C'est pourquoi nous nous proposons de comptabiliser le nombre d'opérations nécessaires à l'obtention d'une solution pour laquelle la norme du résidu est inférieure ou égale à 10^{-16} d'une part et inférieure ou égale à 10^{-8} d'autre part.

De plus, pour chaque méthode, il est indiqué, à titre indicatif, l'itération pour laquelle le critère d'arrêt a été obtenu.

Ce nombre d'opérations a été donné, dans chaque cas, à l'aide de la fonction *flops* disponible sous *Matlab* et comprend les opérations contenues dans l'initialisation de chaque algorithme ainsi que les opérations effectuées jusqu'à l'itération indiquée.

Ainsi, il serait intéressant de pouvoir étudier ces trois méthodes (TFM–Lanczos/Orthomin, TFM–BiCGStab/Orthomin et *Block Bi-CG*) en fonction du nombre de seconds membres considérés. C'est pourquoi nous fournissons ci-après huit tableaux représentant successivement le nombre d'opérations nécessaires pour un second membre à 2, 5, 25 puis 50 colonnes avec un test d'arrêt de 10^{-16} et 10^{-8} (valeur de ε). Dans les tableaux suivants figurent ainsi les résultats obtenus pour chaque algorithme. Pour des raisons évidentes de place, le nom de l'algorithme TFM–Lanczos/Orthomin est abrégé dans le tableau en TFM–L/Omin tandis que celui du TFM–BiCGStab/Orthomin a été abrégé en TFM–BiCGStab.

Les six matrices $\boldsymbol{M}_1$, $\boldsymbol{M}_2$, $\boldsymbol{M}_3$, $\boldsymbol{M}_4$, $\boldsymbol{M}_5$ et $\boldsymbol{M}_6$ que nous avons considéré

correspondent aux six matrices de la sous-section 3.1, sauf que la dimension de chaque matrice est ici ramené à 200.

Un "–" dans une case signifie que le résultat escompté n'a pas été atteint avant 400 itérations, soit deux fois la dimension de la matrice étudiée!

Les nombres en caractères gras dans chaque tableau représentent le nombre d'opérations minimum constaté parmi les trois méthodes testées. Tandis que les nombres en italiques indiquent que l'itération théorique maximale dans chaque méthode a été dépassée pour la matrice considérée.
Le conditionnement de chaque matrice est rappelé, à titre indicatif également, dans la colonne notée *cond.*

L'algorithme de mise en œuvre du *Block Bi-CG* est bien évidemment l'Algorithme 2.1 introduit à la Section 2.1. Pour cet algorithme, les matrices auxiliaires M, γ_k, $\bar{\gamma}_k$ ont toutes été remplacées par des matrices identités de dimensions correspondantes.

3.2.1 Comparaison pour 2 seconds membres

Considérons tout d'abord ce qu'il en est pour 2 seconds membres. Les Tableaux 28 et 29 relatent les données recueillies pour $\varepsilon = 10^{-16}$ et $\varepsilon = 10^{-8}$ respectivement.

$\varepsilon = 10^{-16}$	Cond.	Block Bi-CG	TFM–L/Omin	TFM–BiCGStab
$\boldsymbol{M}_1$ Itération	1.2222	15	13	14
Flops		7 714 962	6 812 776	**4 705 900**
$\boldsymbol{M}_2$ Itération	2.8443	12	11	36
Flops		6 202 301	**5 815 926**	11 763 115
$\boldsymbol{M}_3$ Itération	200	91	152	117
Flops		46 037 784	76 099 245	**39 320 753**
$\boldsymbol{M}_4$ Itération	9.58×10^{13}	–	–	135
Flops		–	–	**45 370 273**
$\boldsymbol{M}_5$ Itération	2.9094	82	93	173
Flops		**41 499 790**	46 689 344	91 751 581
$\boldsymbol{M}_6$ Itération	458.8501	25	57	36
Flops		12 757 415	28 744 600	**12 099 148**

TAB. 28: *Nombre d'opérations nécessaires pour un résidu* $\leqslant 10^{-16}$ *pour les méthodes du Block Bi-CG, TFM–Lanczos/Orthomin et TFM–BiCGStab/Orthomin (2 seconds membres).*

Pour $\varepsilon = 10^{-16}$:

On remarque que, pour 2 seconds membres, la méthode généralement la plus efficace en termes d'opérations effectuées pour les matrices considérées est le TFM–BiCGStab/Orthomin.

Ainsi, même si pour $\boldsymbol{M}_1$ le nombre d'itérations est supérieur à celui requis pour TFM–Lanczos/Orthomin (respectivement 14 contre 13). L'algorithme TFM–BiCGStab/Orthomin requiert moins d'opérations par itération que TFM–Lanczos/Orthomin.

Pour $\boldsymbol{M}_2$, les algorithmes *Block Bi-CG* et TFM–Lanczos/Orthomin sont assez proches, aussi bien du point de vue des itérations que du point de vue du coût opératoire.

$\boldsymbol{M}_3$ nous donne des résultats proches pour le *Block Bi-CG* et le TFM–BiCGStab/Orthomin. Toutefois, le nombre d'itérations pour le *Block Bi-CG* est 91 contre 117 pour le TFM–BiCGStab/Orthomin alors que le nombre d'opérations nécessaires est moindre pour la seconde méthode. Le coût opératoire

$\varepsilon = 10^{-8}$		Cond.	Block Bi-CG	TFM–L/Omin	TFM–BiCGStab
M_1	Itération	1.2222	9	7	8
	Flops		4 689 445	3 913 594	**2 689 471**
M_2	Itération	2.8443	10	9	19
	Flops		5 193 895	**4 808 044**	6 386 065
M_3	Itération	200	69	82	80
	Flops		34 943 923	41 107 909	**26 886 287**
M_4	Itération	9.58×10^{13}	*107*	–	132
	Flops		54 105 627	–	**44 362 134**
M_5	Itération	2.9094	51	50	138
	Flops		25 897 974	**25 195 624**	46 378 376
M_6	Itération	458.8501	15	35	22
	Flops		7 714 966	17 736 647	**7 394 415**

TAB. 29: *Nombre d'opérations nécessaires pour un résidu* $\leqslant 10^{-8}$ *pour les méthodes du Block Bi-CG, TFM–Lanczos/Orthomin et TFM–BiCGStab/Orthomin (2 seconds membres).*

du *Block Bi-CG* est plus élevé par itération (ceci est sûrement dû au calcul de l'inverse à chaque étape, même si, ici, il ne s'agit que d'inverser une matrice de dimension 2).

La matrice M_4 favorise, à nouveau, le TFM–BiCGStab/Orthomin dans la mesure où les deux autres méthodes n'ont pas convergé vers la solution du système. On peut penser que le M-BiCGstab est moins sensible que TFM–Lanczos/Orthomin au conditionnement de la matrice considérée (en effet, ce dernier est relativement élevé pour M_4).

La matrice M_5, quant à elle, est favorable au *Block Bi-CG*, même si le TFM–Lanczos/Orthomin donne des résultats du même ordre (mais toutefois moins performants).

Enfin, pour la matrice M_6, en dépit d'un conditionnement plus important que dans les autres cas, les résultats sont, pour les trois algorithmes, corrects, avec toutefois un léger avantage au TFM–BiCGStab/Orthomin juste devant le *Block Bi-CG*.

Pour $\varepsilon = 10^{-8}$:

Les comportements sont globalement identiques lorsque la précision désirée $\varepsilon = 10^{-16}$. Toutefois, pour M_5, une précision de 10^{-8} entraîne un nombre d'opérations moindre pour TFM–Lanczos/Orthomin alors que pour 10^{-16} le *Block Bi-CG* était le plus performant.

On remarque également que pour $\boldsymbol{M}_4$ la précision 10^{-16} n'était jamais atteinte par le *Block Bi-CG* alors que pour 10^{-8} cet algorithme satisfait au critère d'arrêt (avec néanmoins 107 itérations, ce qui est supérieur au critère théorique de 100 itérations). Toujours pour cette matrice, TFM–Lanczos/Orthomin ne converge pas même si le test d'arrêt est rabaissé à 10^{-8}.

3.2.2 Comparaison pour 5 seconds membres

Nous allons maintenant étudier le comportement des trois méthodes pour 5 seconds membres. Les Tableaux 30 et 31 contiennent les résultats obtenus.

$\varepsilon = 10^{-16}$	Cond.	Block Bi-CG	TFM–L/Omin	TFM–BiCGStab
$\boldsymbol{M}_1$ Itération	1.2222	14	13	16
Flops		19 028 587	10 371 580	**9 431 809**
$\boldsymbol{M}_2$ Itération	2.8443	10	11	32
Flops		13 689 363	**8 865 900**	18 860 299
$\boldsymbol{M}_3$ Itération	200	*65*	154	124
Flops		87 103 421	116 527 410	**73 074 502**
$\boldsymbol{M}_4$ Itération	9.58×10^{13}	–	–	138
Flops		–	–	**81 324 445**
$\boldsymbol{M}_5$ Itération	2.9094	*56*	94	*278*
Flops		75 090 491	**71 354 188**	163 827 915
$\boldsymbol{M}_6$ Itération	458.8501	–	57	36
Flops		–	43 497 664	**21 217 291**

TAB. 30: *Nombre d'opérations nécessaires pour un résidu* $\leqslant 10^{-16}$ *pour les méthodes du Block Bi-CG, TFM–Lanczos/Orthomin et TFM–BiCGStab/Orthomin (5 seconds membres).*

Pour $\varepsilon = 10^{-16}$:

Comme dans le cas de deux seconds membres, la méthode qui semble la plus efficace est le TFM–BiCGSTab/Orthomin.

Ainsi, pour la matrice $\boldsymbol{M}_1$, le TFM–BiCGStab/Orthomin donne de meilleurs résultats que les deux autres algorithmes, même si TFM–Lanczos/Orthomin converge en un nombre d'itérations moindre et en un nombre d'opérations assez proche.

La matrice $\boldsymbol{M}_2$ favorise quant à elle TFM–Lanczos/Orthomin alors que l'algorithme du TFM–BiCGStab/Orthomin a requis plus du double d'opérations pour un résultat analogue. Le *Block Bi-CG*, même avec un nombre d'itérations proche de TFM–Lanczos/Orthomin (10 contre 11 respectivement) nécessite lui aussi beaucoup plus de calculs.

La matrice $\boldsymbol{M}_3$, avec un conditionnement de 200 fait échec au *Block Bi-CG* puisque le critère d'itérations maximales est dépassé (65 alors que 40 sont théoriquement nécessaires). Les deux autres algorithmes convergent assez lentement et le nombre de calculs effectués est le moins élevé pour le TFM–BiCGStab/Orthomin.

$\varepsilon = 10^{-8}$		Cond.	Block Bi-CG	TFM–L/Omin	TFM–BiCGStab
M_1	Itération	1.2222	9	7	9
	Flops		12 354 460	5 854 308	**5 306 842**
M_2	Itération	2.8443	10	9	21
	Flops		13 689 474	**7 359 988**	12 378 300
M_3	Itération	200	*41*	88	83
	Flops		55 067 554	66 836 784	**48 913 634**
M_4	Itération	9.58×10^{13}	–	–	130
	Flops		–	–	**76 610 425**
M_5	Itération	2.9094	37	49	141
	Flops		49 728 932	**37 474 632**	83 092 688
M_6	Itération	458.8501	–	36	28
	Flops		–	27 687 421	**16 503 040**

TAB. 31: *Nombre d'opérations nécessaires pour un résidu* $\leqslant 10^{-8}$ *pour les méthodes du Block Bi-CG, TFM–Lanczos/Orthomin et TFM–BiCGStab/Orthomin (5 seconds membres).*

M_4 a causé des problèmes de stabilité aux deux algorithmes *Block Bi-CG* et TFM–Lanczos/Orthomin. De telle sorte que ces deux méthodes n'ont pas convergé alors que, comme dans le cas où deux seconds membres étaient considérés, le TFM-BiCGStab/Orthomin donne un résultat correct.

La matrice M_5, en dépit d'un conditionnement très bon (2.9094) rend l'algorithme TFM–BiCGStab/Orthomin instable puisque 278 itérations ont été nécessaires à la convergence de l'algorithme. De même, le *Block Bi-CG* dépasse son itération maximale de 40 (puisque la convergence est atteinte en 56 itérations). Toutefois, même si TFM–Lanczos/Orthomin donne les meilleurs résultats, le *Block Bi-CG* a occasionné un nombre d'opérations assez proche.

La dernière matrice, M_6, occasionne à nouveau une défaillance de l'algorithme *Block Bi-CG* et le TFM–BiCGStab/Orthomin nécessite deux fois moins d'opérations que TFM–Lanczos/Orthomin. La convergence a été obtenue en 36 itérations pour le TFM–BiCGStab/Orthomin en dépit d'un conditionnement assez élevé.

Pour $\varepsilon = 10^{-8}$:

Le comportement général des méthodes est à nouveau identique à la précision 10^{-16}. Le nombre maximal théorique d'itérations n'est toutefois plus dépassé pour M_3 et M_5.

3.2.3 Comparaison pour 25 seconds membres

Dans les Tableaux 32 et 33 figurent les données obtenues lorsque 25 seconds membres sont considérés.

$\varepsilon = 10^{-16}$	Cond.	Block Bi-CG	TFM–L/Omin	TFM–BiCGStab
$\boldsymbol{M}_1$ Itération	1.2222	8	13	17
$\boldsymbol{M}_1$ Flops		75 379 136	**29 401 460**	38 734 777
$\boldsymbol{M}_2$ Itération	2.8443	*10*	11	50
$\boldsymbol{M}_2$ Flops		94 135 905	**29 401 460**	113 888 385
$\boldsymbol{M}_3$ Itération	200	–	154	120
$\boldsymbol{M}_3$ Flops		–	382 236 470	**273 305 522**
$\boldsymbol{M}_4$ Itération	9.58×10^{13}	–	–	135
$\boldsymbol{M}_4$ Flops		–	–	**307 466 221**
$\boldsymbol{M}_5$ Itération	2.9094	–	96	*274*
$\boldsymbol{M}_5$ Flops		–	**239 127 928**	624 026 343
$\boldsymbol{M}_6$ Itération	458.8501	–	57	37
$\boldsymbol{M}_6$ Flops		–	142 900 224	**84 282 263**

TAB. 32: *Nombre d'opérations nécessaires pour un résidu* $\leqslant 10^{-16}$ *pour les méthodes du Block Bi-CG, TFM–Lanczos/Orthomin et TFM–BiCGStab/Orthomin (25 seconds membres).*

Pour $\varepsilon = 10^{-16}$:

Pour 25 seconds membres, les résultats sont plus mitigés que pour 2 et 5 seconds membres. En effet, le TFM–BiCGStab/Orthomin et TFM–Lanczos/Orthomin donnent tous deux les meilleurs résultats dans 3 cas sur 6.

Pour la matrice $\boldsymbol{M}_1$, TFM–Lanczos/Orthomin requiert le moins d'opérations, même si en terme d'itérations, le *Block Bi-CG* est plus performant (mais cela est théoriquement normal puisque l'algorithme *Block Bi-CG* est censé donner le résultat théorique exact en $\lfloor n/s \rfloor$ itérations contre n pour TFM–Lanczos/Orthomin).

La matrice $\boldsymbol{M}_2$, elle aussi, fait ressortir TFM–Lanczos/Orthomin, tandis que le TFM–BiCGStab/Orthomin nécessite beaucoup plus d'opérations et d'itérations pour un résultat identique. Le *Block Bi-CG*, à nouveau, converge en un très petit nombre d'itérations (10) mais le nombre d'opérations effectuées est, de loin, le plus important et est de plus supérieur au nombre théorique puisque 8 devraient en effet suffire.

La matrice $\boldsymbol{M}_3$ fait échec au *Block Bi-CG* alors que pour 2 et 5 seconds membres la convergence était effective (mais difficile). Le nombre de seconds

$\varepsilon = 10^{-8}$		Cond.	Block Bi-CG	TFM–L/Omin	TFM–BiCGStab
M_1	Itération	1.2222	8	7	8
	Flops		75 378 751	19 459 068	**18 238 414**
M_2	Itération	2.8443	8	9	18
	Flops		75 379 412	**24 372 948**	41 012 125
M_3	Itération	200	–	85	80
	Flops		–	211 102 824	**182 209 910**
M_4	Itération	9.58×10^{13}	–	–	113
	Flops		–	–	**257 363 599**
M_5	Itération	2.9094	8	49	146
	Flops		**75 377 995**	112 651 592	332 517 544
M_6	Itération	458.8501	–	35	36
	Flops		–	88 254 141	**59 231 160**

TAB. 33: *Nombre d'opérations nécessaires pour un résidu* $\leqslant 10^{-8}$ *pour les méthodes du Block Bi-CG, TFM–Lanczos/Orthomin et TFM–BiCGStab/Orthomin (25 seconds membres).*

membres semble, pour cette matrice, déterminant. Pour les deux autres méthodes, c'est le TFM–BiCGStab/Orthomin qui converge en moins d'itérations et qui nécessite le moins d'opérations.

La matrice M_4, qui causait déjà des problèmes au *Block Bi-CG* et à TFM–Lanczos/Orthomin pour 2 puis 5 seconds membres, ne permet pas, à nouveau, la convergence de ces deux algorithmes pour 25. Le TFM–BiCGStab/Orthomin continue à donner des résultats corrects pour cette matrice. La nature de la matrice semble ici déterminante.

La matrice M_5, comme dans le cas de 5 seconds membres, semble causer des problèmes aux trois algorithmes. De telle sorte que le *Block Bi-CG* ne converge pas et que le TFM–BiCGStab/Orthomin dépasse les 200 itérations. TFM–Lanczos/Orthomin semble, dans ce cas un assez bon algorithme.

Enfin, la matrice M_6, pour laquelle le *Block Bi-CG* ne convergeait pas pour 5 seconds membres, donne les mêmes caractéristiques avec un très bon résultat pour le TFM–BiCGStab/Orthomin.

Pour $\varepsilon = 10^{-8}$:

Des différences avec un test d'arrêt à 10^{-16} existent pour la matrice M_1 ainsi que pour la matrice M_5. Pour la première, TFM–BiCGStab/Orthomin donne de meilleurs résultats que TFM–Lanczos/orthomin si $\varepsilon = 10^{-8}$ alors que c'était le contraire pour $\varepsilon = 10^{-16}$.

Pour la seconde, le *Block Bi-CG* est la plus performante des trois méthodes (en atteignant son nombre maximal théorique d'itérations, 8) si $\varepsilon = 10^{-8}$ alors que dans le cas d'un test d'arrêt à 10^{-16} cette méthode ne converge pas !

3.2.4 Comparaison pour 50 seconds membres

Enfin, dans les Tableaux 34 et 35 figurent les informations obtenues lorsque l'on considère un système linéaire avec 50 seconds membres.

$\varepsilon = 10^{-16}$	Cond.	Block Bi-CG	TFM–L/Omin	TFM–BiCGStab
M_1 Itération	1.2222	*10*	13	15
M_1 Flops		263 186 463	**64 252 840**	65 851 981
M_2 Itération	2.8443	–	11	49
M_2 Flops		–	**55 037 910**	215 026 876
M_3 Itération	200	–	154	128
M_3 Flops		–	713 910 795	**561 640 335**
M_4 Itération	9.58×10^{13}	–	–	135
M_4 Flops		–	–	**592 353 121**
M_5 Itération	2.9094	–	96	*274*
M_5 Flops		–	**446 675 003**	1.2022×10^9
M_6 Itération	458.8501	–	57	36
M_6 Flops		–	266 982 424	**157 989 440**

TAB. 34: *Nombre d'opérations nécessaires pour un résidu* $\leqslant 10^{-16}$ *pour les méthodes du Block Bi-CG, TFM-Lanczos/Orthomin et TFM-BiCGStab/Orthomin (50 seconds membres).*

Pour $\varepsilon = 10^{-16}$:

Comme dans le cas de 25 seconds membres, trois matrices sont favorables à TFM–Lanczos/Orthomin et trois le sont au TFM–BiCGStab/Orthomin et il s'agit des mêmes pour chaque algorithme lorsque le nombre de seconds membres est 50.

Pour la matrice M_1, TFM–Lanczos/Orthomin donne les meilleurs résultats mais le TFM–BiCGStab/Orthomin reste assez proche aussi bien pour le critère des itérations que pour le critère du nombre d'opérations effectuées. Certes le *Block Bi-CG* requiert moins d'itérations mais le nombre de calculs est beaucoup plus important dans cet exemple.

La matrice M_2 est très favorable à TFM–Lanczos/Orthomin alors que cette fois-ci le *Block Bi-CG* n'a pas convergé vers la solution du système considéré. Le fait d'augmenter le nombre de seconds membres en est sûrement la cause (il faut en effet considérer l'inversion de matrices de dimension 50).

La matrice M_3 qui, pour 25 seconds membres, causait une divergence de l'algorithme *Block Bi-CG* donne les mêmes résultats pour 50. Alors que TFM–

$\varepsilon = 10^{-8}$	Cond.	Block Bi-CG	TFM–L/Omin	TFM–BiCGStab
M_1 Itération	1.2222	4	7	8
Flops		103 330 802	36 465 018	**35 139 439**
M_2 Itération	2.8443	4	9	19
Flops		103 330 666	**45 639 148**	83 401 873
M_3 Itération	200	–	82	89
Flops		–	**380 497 329**	390 527 188
M_4 Itération	9.58×10^{13}	–	–	115
Flops		–	–	**504 602 695**
M_5 Itération	2.9094	7	53	138
Flops		**183 256 175**	247 471 363	605 515 544
M_6 Itération	458.8501	–	37	26
Flops		–	174 077 721	**114 114 435**

TAB. 35: *Nombre d'opérations nécessaires pour un résidu* $\leqslant 10^{-8}$ *pour les méthodes du Block Bi-CG, TFM-Lanczos/Orthomin et TFM-BiCGStab/Orthomin (50 seconds membres).*

Lanczos/Orthomin reste la meilleure méthode pour cette matrice (4 fois moins d'opérations que le TFM–BiCGStab/Orthomin).

Depuis 2 seconds membres le *Block Bi-CG* et TFM–Lanczos/Orthomin ne convergent pas pour la matrice M_4 et il en est de même ici pour 50. Le TFM–BiCGStab/Orthomin est le seul à assurer une convergence satisfaisante.

Pour la matrice M_5, comme pour 25 seconds membres, le *Block Bi-CG* ne converge pas alors que le TFM–BiCGStab/Orthomin converge en un nombre d'itérations trop important (274) et un nombre d'opérations très élevé. TFM–Lanczos/Orthomin semble le meilleur des trois algorithmes pour cette matrice.

Le comportement des trois méthodes pour 50 seconds membres est le même que pour 25 en ce qui concerne la matrice M_6. C'est-à-dire que le *Block Bi-CG* ne converge pas et les meilleurs résultats sont obtenus pour le TFM–BiCGStab/Orthomin.

Pour $\varepsilon = 10^{-8}$:

Pour les matrices M_2, M_4 et M_6 les comportements restent les mêmes.

Pour la matrice M_1 TFM–Lanczos/Orthomin donne un résultat légèrement meilleur que TFM–BiCGStab/Orthomin pour $\varepsilon = 10^{-16}$ alors que c'est l'inverse pour $\varepsilon = 10^{-8}$.

Pour M_3, la même remarque peut être formulée avec toutefois un nombre d'opérations bien moindre pour TFM–BiCGStab/Orthomin si $\varepsilon = 10^{-16}$.

Pour M_5, alors que pour un test d'arrêt de 10^{-16} la convergence n'est pas assurée avec le *Block Bi-CG* il en est autrement si $\varepsilon = 10^{-8}$ et c'est alors le meilleur résultat obtenu (même si le nombre maximal théorique d'itérations est dépassé, 7 au lieu de 4).

3.3 Étude de quelques matrices creuses de grandes dimensions

Dans cette sous-section, nous allons traiter des matrices creuses de grandes dimensions afin de voir l'incidence de ces dernières sur la convergence (et aussi la stabilité) des algorithmes TFM–Lanczos/Orthomin et TFM-BiCGStab/Orthomin (puisque l'on a vu que ces deux algorithmes étaient les deux plus efficaces parmi ceux présentés dans les Sections précédentes).

Pour cela, nous allons à nouveau étudier les matrices M_1, M_2, M_3, M_4 et M_5 de la sous-section 3.1, sauf qu'au lieu de les considérer de dimension 500, nous allons travailler sur des matrices de dimension 10 000 (la matrice M_6, même si elle comporte beaucoup de zéros, ne se prête pas très bien à un stockage en matrice creuse).

Les algorithmes TFM–Lanczos/Orthomin et TFM-BiCGStab/Orthomin ont été légèrement modifiés afin d' être compatibles avec un stockage de matrices creuses (changement des produits matrice-vecteur uniquement). Le critère d'arrêt retenu est le même que les critères précédents et concerne la moyenne des normes euclidiennes des résidus.

Le nombre d'opérations pour l'obtention du résidu satisfaisant ne figure pas dans le tableau car il peut dépendre de la façon de stocker les matrices creuses.

Les résultats concernant ces matrices sont donnés dans le tableau 36 de la page suivante pour 5 seconds membres uniquement (on a en effet remarqué que le nombre de seconds membres influait peu la convergence des deux algorithmes en question).

La colonne intitulée "nz" contient tout naturellement le nombre d'éléments non nuls de chaque matrice. Les deux colonnes suivantes concernent respectivement l'algorithme TFM–Lanczos/Orthomin et TFM–BiCGStab/Orthomin.

Les nombres en caractères gras sont mis en évidence car ils représentent, pour une matrice donnée, le nombre d'itérations le plus bas qui a été nécessaire pour obtenir un résidu satisfaisant le critère d'arrêt.

Un '–' dans une case du tableau signifie que le critère d'arrêt n'a pas été atteint pour l' algorithme et la matrice considérée.

On remarque que, pour ces exemples, le comportement des deux algorithmes est semblable à ce qu'il était lorsque les dimensions des matrices étaient plus petites (dans les deux sous-sections précédentes). L'avantage, notamment, d'un

Matrice	nz	TFM–Lanczos/Orthomin	TFM–BiCGStab/Orthomin
M_1	29 998	**15**	17
M_2	29 960	**33**	41
M_3	10 000	1006	**896**
M_4	37 980	–	145
M_5	29 997	**97**	234

TAB. 36: *Nombre d'itérations nécessaires à la résolution de systèmes linéaires où la dimension de la matrice considérée est* 10 000 *pour les méthodes TFM–Lanczos/Orthomin et TFM–BiCGStab/Orthomin pour les matrices* M_1, M_2, M_3, M_4 *et* M_5 *avec* 5 *seconds membres.*

algorithme par rapport à un autre reste, en particulier, identique pour chaque matrice.

C'est-à-dire que l'algorithme TFM–Lanczos/Orthomin donne de meilleurs résultats pour les matrices M_1, M_2 et M_5 tandis que pour les matrices M_3 et M_4, TFM–BiCGStab est le plus performant.

Pour M_4, plus précisément, TFM–Lanczos/Orthomin ne converge toujours pas, comme c'était déjà le cas pour cette matrice lorsque la dimension considérée était 200 ou 500.

Ces exemples concernant les matrices creuses de grandes dimensions correspondant aux matrices des deux Sections précédentes confirment donc que la convergence des deux algorithmes utilisés ne dépend pas de la dimension des matrices utilisées.

En fait, cette convergence semble dépendre exclusivement du conditionnement des matrices (en supposant que le conditionnement de ces cinq matrices évolue proportionnellement à la dimension de celles-ci, ce qui n'est pas nécessairement le cas).

Conclusion

Deux aspects doivent être ici considérés : le point de vue théorique et le point de vue numérique.

Pour le premier critère, nous avons obtenu une interprétation matricielle unique, c'est-à-dire où une seule matrice apparaît pour le calcul des polynômes orthogonaux, en dépit de plusieurs seconds membres (ce qui permet de classer ces méthodes dans les méthodes à "système central"). De plus, une interprétation polynomiale de la méthode, en termes de polynômes biorthogonaux a également été démontrée.

D'autre part, en étudiant les exemples de la sous-section 3.1, ceux de la sous-section 3.2 et ceux de la sous-section 3.3, plusieurs remarques peuvent être formulées.

Premièrement, les algorithmes M–Lanczos/Orthomin et M–Lanczos/Orthodir ne donnent pas des résultats très encourageants en dépit de propriétés théoriques intéressantes. L'algorithme TFM–Lanczos/Orthodir ne donne pas une convergence efficace non plus. Cela doit être dû, en partie, au fait que la suite $\left\{\boldsymbol{x}_k^{(j)}\right\}_{k\geqslant 0}$ est la même pour M–Lanczos/Orthodir et TFM–Lanczos/Orthodir (la différence est la manière de la calculer). Ainsi, si l'un des algorithmes est défaillant, l'autre le sera également et de la même manière.

Deuxièmement, le TFM–BiCGStab/Orthomin et le TFM–Lanczos/Orthomin nous donnent de meilleurs résultats avec des matrices symétriques, même si les matrices non symétriques donnent un comportement correct pour ces deux algorithmes également. Dans les exemples de la Section 3, on ne peut pas dire si l'un des deux algorithmes est meilleur que l'autre. D'un point de vue emplacement mémoire, le TFM–Lanczos/Orthomin nécessite un vecteur supplémentaire et d'un point de vue coût, le TFM–BiCGStab/Orthomin nécessite beaucoup plus de produits scalaires par itération (on retrouve ce coût supplémentaire dans les exemples de la sous-section 3.2).

Toutefois, il semblerait que sur des matrices à conditionnement "faible", l'algorithme TFM–Lanczos/Orthomin donne de meilleurs résultats que TFM–BiCGStab (indépendamment de la dimension des matrices considérées) : c'est notamment le cas des matrices $\boldsymbol{M}_1$, $\boldsymbol{M}_2$ et $\boldsymbol{M}_5$.

Tandis que pour les matrices à conditionnement "élevé", TFM–BiCGStab soit plus efficace : c'est notamment le cas pour les matrices $\boldsymbol{M}_3$, $\boldsymbol{M}_4$ et $\boldsymbol{M}_6$. La notion de grandeur du conditionnement reste malgré tout très subjective.

Troisièmement, le nombre de seconds membres ne paraît pas avoir une influence significative sur la convergence du TFM–BiCGStab/Orthomin dans les exemples étudiés. (sauf, bien entendu, pour le coût opératoire et la place mémoire requise). Il en est de même pour l'algorithme TFM–Lanczos/Orthomin. Cela peut être vérifié aussi bien dans la sous-section 3.1 que dans la sous-section 3.2. Même si le coefficient a_k, pour l'algorithme TFM–BiCGStab/Orthomin, est calculé à partir des différents seconds membres, ceux-ci ne semblent pas être déterminants pour la convergence de l'algorithme.

En outre, même avec la propriété de minimisation du TFM–

BiCGStab/Orthomin, cela ne semble pas être un critère d'accélération de la convergence (puisqu'alors, les résultats devraient être bien meilleurs que le TFM–Lanczos/Orthomin, ce qui n'est pas, en général, le cas, sauf pour M_4 mais cela serait plus probablement dû au conditionnement de cette matrice). Cette constatation est, comme l'a fait remarquer Simoncini (communication personnelle), déjà vraie pour le BiCGStab pour la résolution de systèmes à second membre unique.

Étant basées sur le calcul de polynômes orthogonaux pour le système central, les algorithmes TFM–Lanczos/Orthomin et TFM–BiCGStab/Orthomin pourraient être plus performant si l'on considérait non plus des bases orthogonales $(\widetilde{P}_k,\ P_k^{(0)})$ mais des bases quasi-orthogonales (voir [2]). Encore faut-il étudier si une telle considération est possible et comment la mettre en œuvre. Cette quasi-orthogonalité s'apparente en effet à une orthogonalité numérique, en quelque sorte. Elle paraît donc préférable du point de vue algorithmique et engendrerait, peut-être, une meilleure stabilité numérique.

De plus, lorsque l'on considère les exemples de la sous-section 3.2, on constate que, d'une manière générale, plus le nombre de seconds membres considérés augmente, plus les méthodes TFM–Lanczos/Orthomin et TFM–BiCGStab/Orthomin sont stables par rapport au *Block Bi-CG*. Ceci paraît normal puisque dans la méthode par bloc on considère l'inversion de matrices de dimension s où s est le nombre de seconds membres, ce qui conduit nécessairement à une difficulté numérique supplémentaire.

L'inconvénient majeur de ces algorithmes, comme on peut le voir notamment avec les graphes de la sous-section 3.1, est que l'on n'a pas, en général, une décroissance des résidus. Bien sûr, comme il est dit dans [30], on peut toujours considérer le résidu le plus faible à chaque itération afin d'avoir des courbes décroissantes mais il ne s'agit alors que d'un artifice. L'on peut également voir si des procédures de lissage (*smoothing*) comme celles récemment introduites par Heyouni et al. [44] ne peuvent pas être appliquées. Des méthodes hybrides peuvent quant à elles également être considérées, comme c'était déjà le cas pour un second membre unique dans un article de Sleijpen et al. [72] pour une adaptation du BiCGStab appelée BiCGStab(l) et mise au point dans [66]. Il faut donc voir s'il n'est pas possible de modifier à nouveau ces algorithmes afin d'obtenir cette décroissance (réelle et non pas uniquement visuelle sur les courbes bien entendu), tout en conservant le caractère fini du processus. En particulier, il nous faut maintenant étudier une amélioration numérique des algorithmes M–Lanczos/Orthodir et M–Lanczos/Orthomin puisqu'ils requièrent un faible coût opératoire par rapport a TFM–Lanczos/Orthomin et TFM–BiCGStab/Orthomin (même si la trans-

posée de la matrice $\boldsymbol{A}$ est utilisée), en utilisant, notamment, des polynômes U_k particulier et non les polynômes $U_k(x) = x^k$.

Toutefois, plusieurs travaux relativement récents tendent à montrer pour diverses méthodes que lorsqu'il y a un problème numérique, quel que soit l'algorithme considéré, ce problème persiste [30], [22]. De tels résultats sont à mettre en parallèle avec les travaux de Sidi [60] puisqu'il y est démontré que beaucoup de méthodes sont, à la base, équivalentes.

Ces méthodes doivent également encore être comparées avec les méthodes de type Lanczos existantes, et en particulier avec le Global Lanczos [59]. Une comparaison avec certaines autres méthodes, non issues de méthodes à second membre unique, telle que celle mise au point par T. F. Chan et al. [26] serait également judicieuse.

Enfin, il faudrait également voir si l'on peut donner des versions QMR et préconditionnées des algorithmes étudiés dans cette Section et ce qu'il adviendrait alors du comportement numérique des algorithmes correspondants.

Conclusion générale

Après avoir décrit certaines propriétés que satisfont les polynômes orthogonaux et orthogonaux de dimension $d > 1$, encore appelés polynômes vectoriellement orthogonaux, nous avons pu généraliser certains des résultats connus sur ces derniers aux polynômes biorthogonaux, en trouvant ainsi de nouvelles relations de récurrence ainsi que de nouvelles relations matricielles.

Ces nouvelles relations ont été rendues possibles grâce à des considérations analogues à celles qui avaient été faites pour les polynômes orthogonaux et orthogonaux de dimension $d > 1$. C'est notamment à travers les racines des polynômes biorthogonaux qu'une majeure partie de ces résultats a pu être mise en évidence.

C'est également en considérant ces racines ainsi que des notions telles que polynômes biorthogonaux à droite et polynômes biorthogonaux à gauche que plusieurs identités de type Christoffel-Darboux ont pu être démontrées, généralisant celle existant pour les polynômes orthogonaux.
D'autre part, on a également démontré, à partir de certaines matrices résolvantes, que sous certaines hypothèses, les polynômes biorthogonaux étaient source de projecteurs orthogonaux matriciels, qui ont été pleinement déterminés.

De plus, on a prouvé les relations qui pouvaient exister entre le calcul des polynômes biorthogonaux et la mise en œuvre de la méthode de bordage. Les cas particuliers des matrices de Hankel et de Toeplitz ont tout particulièrement été étudiés.

Tout ceci a donc permis d'enrichir et d'étayer, sous un aspect nouveau, la connaissance, déjà étendue, de ces polynômes biorthogonaux.

Après avoir rappelé certaines des méthodes de résolution de systèmes linéaires à seconds membres multiples et en avoir analysé les principales caractéristiques, le deuxième aspect de cette thèse a également permis, à partir de la méthode de Lanczos et de considérations sur les polynômes biorthogonaux, d'établir une méthode de résolution des systèmes linéaires à seconds membres multiples qui possède des propriétés intéressantes.

Cette méthode présente notamment une interprétation matricielle où une seule matrice est considérée, ce qui, par rapport aux autres méthodes existantes, semble nouveau.

Les algorithmes issus de cette méthode, dont deux présentaient des caractéristiques numériques intéressantes, se sont avérés plus efficaces dans la majeure partie des cas étudiés que la méthode du *Block Bi-CG* tout en nécessitant, en général, moins d'opérations arithmétiques pour l'obtention de la solution aux problèmes posés.

Cependant, l'étude de cette méthode n'est pas terminée dans le sens où aucune majoration ou propriété de décroissance des résidus n'a pu être démontrée. Notre prochain travail essaiera de résoudre ces problèmes.

Références

[1] E. AYACHOUR, Avoiding the look-ahead in the Lanczos method, Note ANO 363, Laboratoire d'Analyse Numérique et d'Optimisation, Université des Sciences et Technologies de Lille, 1996.

[2] B. BECKERMANN, The stable computation of formal orthogonal polynomials, Numer. Algorithms, 11 (1996), 1–23.

[3] W. E. BOYSE, A. A. SEIDL, A block QMR method for computing multiple simultaneous solutions to complex symmetric systems, SIAM J. Sci. Comput., 17 (1996), 263–274.

[4] C. BREZINSKI, *Padé-Type Approximation and General Orthogonal Polynomials*, ISNM vol. 50, Birkhäuser, Basel, 1980.

[5] C. BREZINSKI, Bordering methods and progressive forms for sequence transformations, Zastosow. Mat., 20 (1990), 435–443.

[6] C. BREZINSKI, CGM : a whole class of Lanczos-type solvers for linear systems, Note ANO 253, Laboratoire d'Analyse Numérique et d'Optimisation, Université des Sciences et Technologies de Lille, November 1991.

[7] C. BREZINSKI, A unified approach to various orthogonalities, Ann. Fac. Sci. Toulouse, Math. 1, 3 (1992), 277–292.

[8] C. BREZINSKI, *Biorthogonality and its Applications to Numerical Analysis*, Marcel Dekker, New York, 1992.

[9] C. BREZINSKI, Biorthogonality and conjugate gradient-type algorithms, *Contributions in Numerical Mathematics*, R. P. Agarwal ed., World Scientific, Singapore, 1993, 55–70.

[10] C. BREZINSKI, Formal orthogonality on an algebraic curve, Ann. Numer. Math., 2 (1995), 21–33.

[11] C. BREZINSKI, M. REDIVO ZAGLIA, A new presentation of orthogonal polynomials with applications to their computation, Numer. Algorithms, 1 (1991), 207–222.

[12] C. BREZINSKI, M. REDIVO-ZAGLIA, *Extrapolation Methods Theory and Practice*, Studies in Computational Mathematics, 2, Amsterdam etc. : North-Holland, ix, 464 p. with floppy disk (1991).

[13] C. BREZINSKI, M. REDIVO ZAGLIA, Breakdowns in the computation of orthogonal polynomials, *Nonlinear Numerical Methods and Rational Approximation II*, A. €uyt ed., Kluwer, Dordrecht, 1994, 49–59.

[14] C. BREZINSKI, M. REDIVO-ZAGLIA, Look-ahead in Bi-CGSTAB and other product-type methods for linear systems, BIT, 35 (1995), 169–201.

[15] C. BREZINSKI, M. REDIVO ZAGLIA, On the zeros of various kinds of orthogonal polynomials, Ann. Numer. Math., 4 (1997), 67–78.

[16] C. BREZINSKI, M. REDIVO ZAGLIA, Transpose-free Lanczos-type algorithms for non symmetric linear systems, Numer. Algorithms, 17 (1998), 67–103.

[17] C. BREZINSKI, H. SADOK, Lanczos type algorithms for solving systems of linear equations, Appl. Numer. Math., 11 (1993), 443–473.

[18] C. BREZINSKI, M. REDIVO ZAGLIA, H. SADOK, A breakdown-free Lanczos type algorithm for solving linear systems. Numer. Math. 63, 1 (1992), 29-38.

[19] C. BREZINSKI, M. REDIVO ZAGLIA, H. SADOK, Breakdowns in the implementation of the Lanczos methods for solving linear systems. J. Comput. Appl. Math., 33 (1997), 31–44.

[20] C. BREZINSKI, M. REDIVO-ZAGLIA, H. SADOK, New look-ahead Lanczos-type algorithms for linear systems, Note ANO 378, Laboratoire d'Analyse Numérique et d'Optimisation, Université des Sciences et Technologies de Lille, Octobre 1997.

[21] M. O. BRISTEAU, J. ERHEL, Augmented conjugate gradient. Application in an iterative process for the solution of scattering problems, 1998, Preprint, 1-24.

[22] P. N. BROWN, A theoretical comparison of the Arnoldi and GMRES algorithms, SIAM J. Sci. Stat. Comput., 12 (1991), 58–78.

[23] A. BULTHEEL, *Laurent Series and their Padé Approximants*, Birkhäuser, Basel, 1987.

[24] A. BULTHEEL, M. VAN BAREL, Formal orthogonal polynomials and Hankel/Toeplitz duality, Numer. Algorithms, 10 (1995), 289–335.

[25] R. H. CHAN, M. K. NG, Fast Reliable Algorithms for Matrices with Structure, ed. T. Kailath and A. Sayed, Preprint.

[26] T. F. CHAN, W. L. WAN, Analysis of projection methods for solving linear systems with multiple right-hand sides. SIAM J. Sci. Comput., 18 (1997), 1698–1721.

[27] T. F. CHAN, E. GALLOPOULOS, V. SIMONCINI, T. SZETO, C. H. TONG, A quasi-minimal residual variant of the Bi-CGSTAB algorithm for nonsymmetric systems, SIAM J. Sci. Comput., 15 (1994), 338–347.

[28] T. F. CHAN, L. DE PILLIS, H. A. VAN DER VORST, Transpose-free formulations of Lanczos-type methods for nonsymmetric linear systems, Numer. Algorithms, 17 (1998), 51–66.

[29] R. E. CLINE, R. J. PLEMMONS, G. WORM, Generalized inverses of certain Toeplitz matrices, Linear Algebra Appl., 8 (1974), 25–33.

[30] J. CULLUM, A. GREENBAUM, Relations between Galerkin and norm-minimizing iterative methods for solving linear systems, SIAM J. Matrix Anal. Appl., 17 (1996), 223–247.

[31] B. W. DICKINSON, An inverse problem for Toeplitz matrices, Linear Algebra Appl., 59 (1984), 79–83.

[32] A. DRAUX, *Polynômes Orthogonaux Formels–Applications.* LNM 974, Springer-Verlag, Berlin, 1983.

[33] A. DRAUX, A. MAANAOUI, Vector orthogonal polynomials, J. Comput. Appl. Math., 32 (1990), 59–68.

[34] FADDEEVA V. N., *Computational Methods of Linear Algebra*, Dover, New York. (1959).

[35] H. LE FERRAND, Vector orthogonal polynomials and matrix series, J. Comput. Appl. Math., 45 (1993), 267–282.

[36] R. FLETCHER, Conjugate gradient methods for indefinite systems, Numer. Anal., Proc. Dundee Conf. 1975, Lect. Notes Math. 506, 73-89 (1976).

[37] R. W. FREUND, N. M. NACHTIGAL, QMR : A quasi-minimal residual method for non- Hermitian linear systems, Numer. Math., 60 (1991), 315–339.

[38] R. W. FREUND, M. MALHOTRA, A block QMR algorithm for non-hermitian linear systems with multiple right-hand sides, Linear Algebra Appl., 254 (1997), 119-157.

[39] P. A. FUHRMANN, Remarks on the inversion of Hankel matrices, Linear Algebra Appl., 81 (1986), 89–104.

[40] L. GEMIGNANI, Fast inversion of Hankel and Toeplitz matrices, Inf. Process. Lett., 41 (1992), 119–123.

[41] W. B. GRAGG, Matrix interpretations and applications of the continued fraction algorithm, Rocky Mountain J. Math., 4 (1974), 213–225.

[42] GRAVES-MORRIS, A "Look–around Lanczos" algorithm for solving a system of linear equations, Numer. Algorithms, 15 (1997), 247–274.

[43] M. H. GUTKNECHT, Z. STRAKOS, Accuracy of the three-term and the two-term recurrences for Krylov space solvers, Technical report 97–21, Swiss Center for Scientific Computing, Zürich, December 1997.

[44] M. HEYOUNI, H. SADOK, On a variable smoothing procedure for Krylov subspace methods, Linear Algebra Appl., 268 (1998), 131–149.

[45] N. J. HIGHAM, The test matrix Toolbox for Matlab (Version 3.0), Numerical analysis report No. 276, Department of Mathematics, The University of Manchester, September 1995.

[46] T. KAILATH, A. VIEIRA, M. MORF, Inverses of Toeplitz operators, innovations, and orthogonal polynomials, SIAM Rev., 20 (1978) 106–119.

[47] C. LANCZOS, An iteration method for the solution of the eigenvalue problem of linear differential and integral operators, J. Res. Nat. Bur Standards, 45 (1950), 255–282.

[48] C. LANCZOS, Solution of systems of linear equations by minimized iterations, J. Res. Natl. Bur. Stand., 49 (1952), 33–53.

[49] A. LASCOUX, Inversion des matrices de Hankel, Linear Algebra Appl., 129 (1990), 77–102.

[50] D. P. O'LEARY, The block conjugate gradient algorithm and related methods, Linear Algebra Appl., 29 (1980), 293–322.

[51] F. MARCELLÀN, L. MORAL, Minimal reccurence formulas for orthogonal polynomials on Bernoulli's lemniscate, *Polynômes Orthogonaux et Applications*, C. Brezinski et al. eds, LNM vol. 1171, Springer-Verlag, Berlin, 1985, 211–220.

[52] P. MARONI, Une généralisation du théorème de Favard-Shohat sur les polynômes orthogonaux, C.R. Acad. Sci., Paris, Ser. I 293, 19-22 (1981).

[53] C. MUSSCHOOT, C. BREZINSKI, Biorthogonal polynomials and the bordering method for linear system, Rend. Semin. Mat. Fis. Milano 64 (1994), 85–98 (1996).

[54] C. MUSSCHOOT, A Lanczos-type method for solving nonsymmetric linear systems with multiple right-hand sides – Matrix and polynomial interpretation, J. Comput. Appl. Math., à paraître.

[55] M.A. PIÑAR, V. RAMIREZ, Recursive inversion of Hankel matrices, Monogr. Acad. Ciencias Zaragoza, 1 (1988) 119–128.

[56] M.A. PIÑAR, V. RAMIREZ, Inversion of Toeplitz matrices, in *Orthogonal Polynomials and their Applications*, J. Vinuesa ed., Marcel Dekker, New York, 1989, 171–177.

[57] J. RISSANEN, Solution of linear equations with Hankel and Toeplitz matrices, Numer. Math., 22 (1974) 361–366.

[58] Y. SAAD, *Iterative Methods for Sparse Linear Systems*, PWS, Boston, 1995.

[59] H. SADOK, K. JBILOU, Global Lanczos-type methods with applications, Applied Linear Algebra, submitted.

[60] A. SIDI, Extrapolation vs. projection methods for linear systems of equations, J. Comput. Appl. Math., 22 (1988), 71–88.

[61] V. SIMONCINI, A stabilized QMR version of block BiCG, SIAM J. Matrix Anal. Appl., 18 (1997), 419–434.

[62] V. SIMONCINI, E. GALLOPOULOS, Convergence properties of block GMRES for solving systems with multiple right-hand sides, Tech. Rep. 1316, Center for Supercomputing Research and Development, University of Illinois at Urbana-Champaign, Oct. 1993.

[63] V. SIMONCINI, E. GALLOPOULOS, An iterative method for nonsymmetric systems with multiple right-hand sides, SIAM J. Sci. Comput., 16 (1995), 917–933.

[64] V. SIMONCINI, E. GALLOPOULOS, A hybrid block GMRES method for nonsymmetric systems with multiple right hand sides, J. Comput. Appl. Math., 66 (1996), 457–469.

[65] V. SIMONCINI, E. GALLOPOULOS, Convergence properties of Block GMRES and matrix polynomial, Linear Algebra Appl., 247 (1996), 97–119.

[66] G. L. G. SLEIJPEN, D. R. FOKKEMA, BiCGStab(l) for linear equations involving matrix complex spectrum, Electron. Trans. Numer. Anal., 1 (1993), 11–32.

[67] C. F. SMITH, A. F. PETERSON, R. MITTRA, A conjugate gradient algorithm for the treatment of multiple incident electromagnetic, IEEE Trans. Antennas and Propagation, 37 (1989), 1490–1493.

[68] P. SONNEVELD, CGS, a fast Lanczos-type solver for nonsymmetric linear systems, SIAM J. Sci. Stat. Comput., 10 (1989), 36–52.

[69] W. TRENCH, An algorithm for the inversion of finite Toeplitz matrices, SIAM J. Appl. Math., 12 (1964) 515–522.

[70] W. TRENCH, An algorithm for the inversion of finite Hankel matrices, SIAM J. Appl. Math., 13 (1965) 1102–1107.

[71] H. A. VAN DER VORST, Bi-CGSTAB : a fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems, SIAM J. Sci. Stat. Comput., 13 (1992), 631–644.

[72] G. L. G. SLEIJPEN, H. A. VAN DER VORST, D. R. FOKKEMA, BiCGSTAB(l) and other hybrid methods, Numer. Algorithms, 7 (1994), 75–109.

[73] J. VAN ISEGHEM, *Approximants de Padé Vectoriels*, Thèse, Université des Sciences et Technologies de Lille Flandres-Artois, 1987.

[74] J. VAN ISEGHEM, Vector orthogonal relations. Vector QD-algorithm, J. Comput. Appl. Math., 19 (1987), 141–150.

[75] J. VAN ISEGHEM, Convergence of the vector QD-Algorithm. Zeros of vector orthogonal polynomials, J. Comput. Appl. Math. 25, 1 (1989), 33–46.

[76] J. VAN ISEGHEM, Rodrigues formula and orthogonality, Note ANO 339, Laboratoire d'Analyse Numérique et d'Optimisation, Université des Sciences et Technologies de Lille, July 1995.

[77] J. VAN ISEGHEM, V. N. SOROKIN, Algebraic aspects of matrix orthogonality for vector polynomials, J. Approx. Theory, 90, 1 (1997) 97–116.

[78] L. VERDE–STAR, Biorthogonal polynomial bases and Vandermonde-like matrices, Stud. Appl. Math., 95 (1995), 269–295.

[79] B. VITAL, *Etude de quelques méthodes de résolution de problèmes linéaires de grande taille sur multiprocesseur*, Thèse, Université de Rennes I, 1990.