

Matrices structurées et matrices de Toeplitz par blocs de Toeplitz en calcul numérique et formel

Houssam Khalil

▶ **To cite this version:**

Houssam Khalil. Matrices structurées et matrices de Toeplitz par blocs de Toeplitz en calcul numérique et formel. Mathématiques [math]. Université Claude Bernard - Lyon I, 2008. Français. NNT : . tel-00306987

HAL Id: tel-00306987
https://theses.hal.science/tel-00306987

Submitted on 28 Jul 2008

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

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

N° d'ordre :135-2008 Année : 2008

THÈSE

présentée devant

l'UNIVERSITÉ CLAUDE BERNARD - LYON 1

pour l'obtention du

DIPLOME DE DOCTORAT

(arrêté du 7 août 2006)

Spécialité : Mathématiques Appliquées

présenté et soutenue publiquement le

25 juillet 2008

par

Houssam KHALIL

Matrices structurées et matrices de Toeplitz par blocs de Toeplitz en calcul numérique et formel

---

Thèse dirigée par : **Michelle Schatzman** et **Bernard Mourrain**

Rapporteurs :

Victor Pan
Marc Van Barel

Membres du jury :

Michelle Schatzman	Directrice de recherche CNRS	Directrice de la thèse
Bernard Mourrain	Directeur de recherche INRIA	Directeur de la thèse
Marc Van Barel	Professeur à Katholieke Universiteit Leuven	Rapporteur
Christophe Sabot	Professeur à l'Université Lyon1	Examinateur

À Hodna, ma chère épouse.

Remerciements

Je tiens à remercier de tout coeur Michelle Schatzman. Je suis fier d'être le dernier de ses étudiants et je resterai admiratif face à l'immensité de ses connaissances et de sa bonhomie, face à son courage et sa sourie permanente contre la maladie. Merci pour tous les efforts dont vous avez fait pour que cette thèse marche. Merci de tout ce que vous m'avez appris. Nos discussions en dehors de maths étaient un vrai plaisir pour moi, espérons que la paix règne à la fin en moyen orient...

J'exprime toute ma gratitude à Bernard Mourrain, mon encadrant à l'INRIA. Merci pour les nombreux séjours à Sophia, pour tes conseils, ta simplicité et ta disponibilité.

Je tiens à remercier Victor Pan et Marc Van Barel de m'avoir fait l'honneur d'accepter d'être rapporteurs de cette thèse, et pour leur patience de lire cette mémoire rédigée en français.

Je remercie également Christophe Sabot d'avoir bien voulu participer au jury de soutenance de ma thèse.

Je remercie tous les membres de l'ICJ et tous les membres de GALAAD. Je cite en particulier, Laurent Busé pour le vif intérêt qu'il a su porter à mes problèmes et à mes interrogations, même à celles qui étaient éloignées de ses domaines de recherche. Thierry Dumont pour son aide en C++ et son aide informatique. Monique pour son aide administratif. Pierre Crepel pour son salut quotidien.

Merci aux thésards que j'ai côtoyé pendant ces années de thèse, je voudrais leur faire part de toute affection pour leur présence et les moments qu'on a pu partager. En particulier, Nader pour ses pauses de café, Amer, Guillaume et Alexandre.

Et parce qu'il n'y avait pas que la thèse mais aussi le monitorat et l'ATER, un merci à Michel Cretin le responsable de la commission formation. À mes étudiants et à tous les personnes avec qui j'ai enseigné.

Et parceque malgré ce travail, j'avais aussi une vie sociale, avec des gens normaux, qui ne comprenaient pas un traître mot de ce que je racontais, j'adresse un merci à mes amis lyonnais, à mes amis de Sophia et partout dans la France, à mes amis au Liban, au canada.....

J'adresse un merci particulier à mes parents et à mes frères, à qui je pense tous les jours et que j'aime profondément, pour les sacrifices dont ils ont fait pour moi, sans lesquels rien de tout cela ne serait arrivé, pour leur soutient permanent. Mille mercis maman.

Je remercie enfin et surtout celle avec qui j'ai la chance de partager ma vie. Elle a été pour moi un soutien constant. Merci à toi Hodna.

Table des matières

Table des notations **7**

1 Introduction **9**

2 Matrices structurées **13**

2.1 Introduction 13

2.2 Préliminaire 15

2.2.1 FFT 15

2.2.2 Matrice circulante 17

2.2.3 Matrices φ-circulantes 19

2.3 Structure de déplacement 20

2.3.1 Opérateurs de déplacement 21

2.3.2 Opérations de base 22

2.3.3 Matrice de type Toeplitz 25

2.3.4 Matrice de type Hankel 26

2.3.5 Matrice de type Vandermonde 27

2.3.6 Matrice de type Cauchy 28

2.3.7 Relation entre les différents types des matrices structurées 29

2.4 Multiplication rapide 32

2.5 Algorithmes rapides de résolution des systèmes structurées 35

2.5.1 Résolution d'un système de Toeplitz 35

2.5.2 Résolution d'un système de Vandermonde 41

2.5.3 Résolution d'un système de Cauchy 41

2.5.4 Résolution d'un système à matrice structurée 42

3 Où trouve-t-on des matrices de Toeplitz par blocs de Toeplitz **45**

3.1 Introduction 45

3.2 EDP et matrices TBT 46
3.3 Résolution de systèmes d'équations polynomiales 55
3.4 Traitement d'images numériques et du signal 56

4 Peut-on trouver des solveurs rapides pour de matrices TBT 59
4.1 Introduction 59
4.2 Matrices de Toeplitz par blocs 61
4.3 Matrices de Toeplitz par blocs de Toeplitz 63
4.3.1 Le déplacement en deux dimensions 64
4.4 Cas particuliers et quelques idées de résolution 68
4.4.1 Matrice circulante par blocs circulants 69
4.4.2 Algorithme de Wiedemann 71
4.4.3 Propriétés tensorielles des matrices de Toeplitz par blocs de Toeplitz 73

5 Toeplitz bande par blocs Toeplitz bande 75
5.1 Introduction 75
5.2 Statistiques pour des matrices bandes Toeplitz et pour des matrices bandes Toeplitz par blocs bande Toeplitz 76
5.2.1 État des connaissances 76
5.2.2 Algorithmes 76
5.2.3 Les résultats dans le cas scalaire 77
5.2.4 Les résultats dans le cas par blocs 77
5.3 Cas scalaire 77
5.3.1 Transformation en matrice circulante plus matrice de petit rang . . 77
5.3.2 Plongement dans une matrice engendrée par $Z + Z^T$ 85
5.3.3 Plogement dans une matrice circulante 88
5.4 Cas par blocs 90
5.4.1 Transformation en matrice circulante plus matrice de petit rang . . 90
5.4.2 Plongement dans une matrice engendrée par $Z + Z^T$ 92
5.4.3 Plogement dans une matrice circulante par blocs circulants 95

6 TBT et syzygies 97
6.1 Introduction 97
6.2 Cas scalaire 98
6.2.1 Syzygies et matrices de Toeplitz 100

6.2.2 Construction des générateurs 104
6.2.3 Division Euclidien 108
6.3 Cas de deux variables 109
6.3.1 Matrices de Toeplitz par blocs de Toeplitz et syzygies 111
6.3.2 Générateurs et réduction 113
6.3.3 Construction des générateurs et division 115

Conclusion et perspectives **117**

Bibliographie **119**

Table des notations

flop	Opération en arithmétique flottante.
$\mathcal{O}(n)$	Notation de Landau.
$\mathbb{K}$	Corps commutatif, le corps de base. En général, il égale $\mathbb{R}$ ou $\mathbb{C}$.
$\mathbb{K}^{m\times n}$	L'espace des matrices à m lignes et n colonnes, à coefficients dans $\mathbb{K}$.
ω_n	$e^{2i\pi/n}$.
$\lfloor x \rfloor$	Le plus grand entier $\leq x$ (partie entière de x).
$\log n$	$\lceil \log_2 n \rceil$.
I_n	La matrice identité de taille $n \times n$.
I	I_n s'il n'y a pas de confusion.
0_n	Le vecteur identiquement nul de longueur n.
e_k	La $k^{\text{ième}}$ colonne de I.
A^{-1}	L'inverse de la matrice A.
A^T	La matrice transposée de A.
A^{-T}	$(A^{-1})^T = (A^T)^{-1}$.
A^*	La transposée conjuguée de A.
$D(v)$	Matrice diagonale avec $v_1, \ldots, v_n$ sur la diagonale.
$L(v)$	Matrice de Toeplitz triangulaire inférieure de première colonne v.

$v \cdot w$	Le produit scalaire de v et w.
$\kappa(A)$	$\|A\|_2 \|A^{-1}\|_2$, nombre de conditionnement de A.

Chapitre 1

Introduction

La résolution des systèmes d'équations linéaires appartient aux problèmes les plus anciens dans les mathématiques et ceux-ci apparaissent dans presque tous les domaines scientifiques. Un algorithme efficace et rapide, pour résoudre un système $Ax = b$, est donc crucial.

En 1750 Cramer a formulé une procédure générale pour résoudre ce système. Cependant, quand on a voulu appliquer cette règle aux nouveaux problèmes en astronomie de cette époque, où le nombre d'équations était grand (10 inconnues), le nombre des opérations arithmétiques qu'on devait faire était énorme (à l'ordre de 400 millions. La méthode de résolution de Cramer est de l'ordre $n(n+1)!$ flops pour un système de taille $n \times n$). Face à ce problème, d'autres algorithmes ont été développés. C'est le cas de la fameuse élimination Gaussienne en 1810.

Les méthodes de résolution basées sur l'élimination de Gauss, qui sont de l'ordre n^3 flops, ont réduit significativement le nombre d'opérations nécessaires pour résoudre un système linéaire. Cependant, plusieurs applications requièrent la résolution de systèmes linéaires de taille $n \times n$ avec n très grand. Dans ces cas, l'élimination de Gauss standard sera très lente. C'est pour cela qu'on cherche à utiliser la structure de la matrice pour réduire le temps de calcul.

Parmi les structures, ce sont les matrices de Toeplitz par blocs de Toeplitz (TBT) qui nous intéressent. Une matrice de Toeplitz par blocs de Toeplitz, de taille $mn \times mn$, formée de m blocs de taille $n \times n$ chacun est une matrice de la forme suivante :

$$T = \begin{pmatrix} T_0 & T_{-1} & \dots & T_{-m+1} \\ T_1 & T_0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & T_{-1} \\ T_{-m+1} & \dots & T_1 & T_0 \end{pmatrix},$$

où, pour chaque i entre $-m+1$ et $m-1$, T_i est une matrice de Toeplitz de taille $n \times n$,

elle est donc de la forme suivante :

$$T_i = \begin{pmatrix} t_{i,0} & t_{i,-1} & \dots & t_{i,-n+1} \\ t_{i,1} & t_{i,0} & \ddots & \vdots \\ \vdots & \ddots & \ddots & t_{i,-1} \\ t_{i,-n+1} & \dots & t_{i,1} & t_{i,0} \end{pmatrix}$$

et plus généralement, nous sommes intéressés par les matrices de Toeplitz multiniveaux données par la définition suivante :

Définition 1.0.1. *Soit E et F deux sous-ensembles de $\mathbb{N}^k$. Soit $T = (T_{\alpha,\beta})_{\alpha\in E,\beta\in F}$ une matrice dont les lignes sont indexées par les éléments de E et les colonnes sont indexées par les éléments de F.*

La matrice T est dite de Toeplitz de niveau k, de taille $card(E) \times card(F)$, si pour tous $\alpha \in E$ et $\beta \in F$, le coefficient $T_{\alpha,\beta} = t_{\alpha-\beta}$ dépend seulement en $\alpha - \beta$.

La résolution des systèmes de Toeplitz multiniveaux n'est pas très étudiée dans la litérature. Par contre, Les matrices de Toeplitz scalaires sont très bien étudiées, et plusieurs algorithmes de résolution rapides, en $\mathcal{O}(n^2)$ flops, et ultra-rapides, en $\mathcal{O}(n \log^2 n)$ flops, sont développés. D'autres matrices scalaires avec structure, comme matrice de Hankel de Vandermonde ou de Cauchy, sont étudiées, et des algorithmes de résolution rapides et ultra-rapides, pour chaque classe, sont donnés. De plus, La notion de rang de déplacement permet d'unifier l'étude de ces structures, et beaucoup d'autres structures, dans une seule approche, en définissant les matrices de type Toeplitz, de type Hankel, de type Vandermonde et de type Cauchy, qui sont des matrices de petit rang de déplacement. Ainsi, la multiplication rapide des matrices structurées par un vecteur, et la résolution, rapide et ultra-rapide, des systèmes structurés, sont possibles.

Si T est une matrice TBT de taille $mn \times mn$, avec n la taille des blocs, alors T est une matrice de type Toeplitz de rang de déplacement vaut $2m$. Plus généralement, si T est une matrice de Toeplitz de niveau k de taille $\prod_{i=1}^{k} n_i$, alors T est de type Toeplitz de rang de déplacement vaut $2\prod_{i=1}^{k-1} n_i$. Donc, en utilisant les algorithmes de résolution des matrices de type Toeplitz, on peut résoudre un système TBT (resp. un système de Toeplitz de niveau k) en $\mathcal{O}(N^3/n^2 \log^2 N)$ (resp. en $\mathcal{O}(N^3/n_k^2 \log^2 N)$) avec $N = mn$ (resp. $N = \prod_{i=1}^{k} n_i$). On peut donc, en utilisant les techniques développées pour le cas scalaire, diviser la complexité par la carrée de la taille de dernier bloc. Ceci dû à l'utilisation de la structure dans une seule direction. Notre objectif est d'utiliser la structure dans toutes les directions et d'arriver, si possible, à un algorithme de résolution en $\mathcal{O}(N \log^\omega N)$ avec $\omega \in \mathbb{R}$.

L'utilisation de la notion de déplacement, et surtout la notion de "petit rang de déplacement" , dans le cas biniveaux n'est pas efficace. La question intéressante sera : est ce qu'une autre théorie que celle de déplacement, doit être développée pour le cas multiniveaux ? On ne répond pas à cette question et on ne démontre non plus la première prétention, par contre on donne quelques réflexions sur ce sujet. L'étude direct des matrices TBT, sans passer par la structure de déplacement est très difficile aussi, parce que les blocs de ces matrices perdent facilement leur structure de Toeplitz.

Cependant, la relation entre les matrices multiniveaux et les polynômes est toujours possible. Ceci donne la possibilité de tirer plusieurs propriétés.

Parmi les propriétés les plus importantes, il y a la multiplication rapide en $\mathcal{O}(N \log N)$ flops pour une matrice T de taille $N \times N$ par un vecteur. Cette propriété fait penser directement aux algorithmes itératives, pour essayer de trouver des algorithmes rapides. Un type d'algorithmes itératives demande, à chaque itération, la multiplication de la matrice T par une autre matrice, ce qui détruit la structure de T, et surtout dans le cas de Toeplitz multiniveaux, et la multiplication devient de plus en plus lente. Donc ce type d'algorithmes, dont l'algorithme de Newton fait partie, n'est pas très intéressant. L'autre type d'algorithmes itératives, demande à chaque itération la multiplication de T par un vecteur, et donc chaque itération demande $\mathcal{O}(N \log N)$ flops. Un tel algorithme sera intéressant s'il converge rapidement. Or la construction des préconditionneurs pour les matrices de Toeplitz multiniveaux n'est pas simple, et ce n'est même pas possible pour plusieurs algèbres de matrices. L'algorithme de Weidemann donne la solution exacte après N itérations ce qui fait un coût total de $\mathcal{O}(N^2 \log N)$ flops.

Une autre propriété des matrices TBT est qu'on peut donner une relation entre la solution du système TBT, $Tx = b$, et les syzygies des polynômes en deux variables. On peut voir cette relation comme une formule de Gohberg-Semencul pour les matrices TBT. Dans le cas scalaire le calcul des syzygies est simple et rapide, et le calcul de la solution à partir de ces syzygies, qui est le reste d'une division euclidienne, est aussi rapide. Par contre dans le cas des matrices TBT, le calcul sera beaucoup plus difficile et il n' y a pas des algorithmes rapides qui calculent les syzygies ni des algorithmes qui calculent rapidement le reste de la division.

Dans le cas particulier où T est une matrice de Toeplitz bande par blocs de Teoplitz bande, de taille $N \times N$ et les bandes sont petites devant $\sqrt{N}$, alors on peut la décomposer en somme d'une matrice circulante et une matrice de petit rang, ceci nous permet de résoudre le système $Tx = b$ en $\mathcal{O}(N^{3/2} \log^2 N)$ flops, qui est dans ce cas plus rapide que l'élimination Gaussien pour les matrices creuses. On peut aboutir au même résultant en plongeant T dans une matrice circulante par blocs circulants (resp. si T est symétrique, dans une matrice de l'algèbre engendré par $Z + Z^T$, où Z est la matrice de shift vers le bas.), de taille un peu plus grand et qui dépend des largeurs des bandes. Mais dans ce cas des matrices de Toeplitz bande par blocs Toeplitz bande, on remarque un comportement bizarre qu'on peut résumer ainsi : si on prend des matrices (pseudo-)aléatoires dans cette classe, une statistique expérimentale montre qu'elles sont mal conditionnées, et donc le système correspondant difficile à résoudre.

Contenu de la thèse

Cette thèse est constituée de cinq chapitres :

Le chapitre suivant est une introduction générale aux matrices structurées scalaires. on rappelle, dans ce chapitre, l'algorithme de transformation de Fourier rapide (FFT), et les propriétés des matrices circulantes. On donne la définition générale du déplacement pour les structures de Toeplitz, de Hankel, de Vandermonde et de Cauchy, et on donne les outils de base qui nous permettent de travailler avec les matrices de déplacement. On étudie aussi

chaque type de matrice structurée, ce qui nous permet de donner des relations entre les différents types de matrices structurées. On décrit ensuite la multiplication rapide d'une matrice structurée par un vecteur. A la fin, on discute les différents types d'algorithmes, rapide et ultra-rapide, de résolution de systèmes structurés.

On donne dans le troisième chapitre des applications, en équations aux dérivées partielles, en géométrie algébrique et en traitement du signal, qui donnent des systèmes TBT. Dans les équations aux dérivées partielles, dès qu'on a un problème non trivial, nous trouverons des matrices TBT, en dimension 2, pour un maillage uniforme et des coefficients constants ; en dimension 3, la généralisation est évidente : matrice de Toeplitz à trois niveaux. Si on tient compte des conditions aux limites dans des cas suffisamment généraux, on devra passer de matrices TBT à des objets plus généraux, par exemple les matrices de type TBT. En géométrie algébrique, les relations entre polynômes en plusieurs variables et matrices structurées multiniveaux sont évidentes. Multiplier deux polynômes en deux variables est équivalent à multiplier une matrice TBT par un vecteur. La résolution des systèmes polynomiaux spéciaux peut se réduire à des systèmes linéaires dont la matrice est Toeplitz multiniveaux. Les matrices TBT sont assez présente dans le domaine de traitement de signal et surtout dans le traitement d'images.

On étudie les matrices TBT et le déplacement en biniveaux dans le quatrième chapitre. On démontre quelques propriétés pour les matrices TBT. Puis, on étudie la difficulté de la généralisation de la notion de déplacement au cas biniveaux, ce qui se traduit par une difficulté de généralisation des algorithmes de résolution des système structurés scalaires aux cas biniveaux.

Dans le cinquième chapitre, on traite le cas particulier des matrices de Toeplitz bandes par Toeplitz bande. On donne pour ce cas de système, deux algorithmes de résolution rapide. On donne une statistique expérimentale qui prouve que ces matrices sont mal conditionnées et que c'est pas le cas pour les matrices de Toeplitz bande scalaires.

Dans le sixième chapitre, on étudie la résolution d'un système de Toeplitz scalaire, $T\,u = g$ d'un nouveau point de vue, en donnant une relation entre la solution d'un tel problème et les syzygies des polynômes d'une seule variable. On donne une connexion explicite entre les générateurs d'une matrice de Toeplitz et les générateurs du module des syzygies correspondant. On démontre que ce module est engendré par deux éléments de degré n et que la solution de $T\,u = g$ peut être interprétée comme le reste d'un vecteur, associé à g, par ces deux générateurs. On fait la généralisation de cette approche aux problèmes en plusieurs variables. On démontre, tout d'abord, comment on peut étendre la notion des générateurs en une matrice TBT T, puis on décrit la structure des générateurs du module des syzygies correspondants aux générateurs de T, et comment on peut déduire la solution du problème $T\,u = g$ à partir de ces générateurs. Ce qui donne un nouveau point de vue pour résoudre les système de Toeplitz par blocs de Toeplitz.

Chapitre 2

Introduction générale aux matrices structurées et résolution rapide pour les matrices scalaires structurées

2.1 Introduction

Plusieurs problèmes en mathématiques appliquées requièrent la résolution de systèmes linéaires de taille $n \times n$. Pour des systèmes de petite taille, il y a pas de grande avantage à utiliser des algorithmes de résolution non standards. Cependant, n peut être très grand, et parfois ces systèmes doivent être résolus de multiples fois. Dans de tels cas, les algorithmes standards basés sur l'élimination de Gauss demandent $\mathcal{O}(n^3)$ opérations arithmétiques pour un système de taille $n \times n$, et ce sera un handicap pour le calcul. C'est pour cela qu'on cherche à utiliser la structure pour réduire le temps de calcul. Les structures de matrices creuses, bandes, triangulaires, symétriques... mais aussi de Toeplitz, de Hankel, de Vandermonde, de Cauchy et beaucoup d'autres types sont sont couramment exploitées.

Rappelons rapidement la définition de ces dernières structures algébriques : une matrice de Toeplitz (resp. de Hankel) est constante le long de ses diagonales descendante (resp. montantes), le terme général d'une matrice de vandermonde est de la forme x_i^j et le terme général d'une matrice de Cauchy est de la forme $1/(s_i - t_j)$.

Les algorithmes de calcul matriciel développés depuis les années cinquante exploitent les différents types de structures creuses, mais aussi les structures de nature plus algébrique.

Les plus connues sont les matrices de Toeplitz et de Hankel, mais les matrices de Vandermonde et de Cauchy sont familière aussi. Ces dernières sont utiles pour leurs applications en mécanique céleste et décodage algébrique. Ces quatre types apparaissent en relation avec les polynômes, l'interpolation rationnelle et l'évaluation multipoint (voir [9], [103], [98]). On conseille au lecteur de se référer à [18], [69], [133], [120], [111], [94] et [47] qui décrivent certaines des nombreuses applications pratiques et théoriques du calcul avec les matrices structurées. On peut définir quatre classes plus générales : type Toeplitz, type Hankel, type Vandermonde, type Cauchy qui incluent les matrices de Bézout, de Loewner, de Pick, de Sylvester et matrices de sous-resultant.

La complexité du calcul avec une matrice structurée de taille $n \times n$ est beaucoup plus faible que pour une matrice générale de taille $n \times n$. En effet, pour une matrice pleine générale il faut n^2 mots en mémoire pour la stocker et $\mathcal{O}(n^\omega)$ opérations avec $2.37 < \omega \leq 3$ pour résoudre le système et pour $\omega < 3$, la constante est très grande. Par contre, les algorithmes les plus rapides pour des matrices pleines structurées demandent $\mathcal{O}(n)$ mots en mémoire et $\mathcal{O}(n \log^2 n)$ opérations pour résoudre le système.

Les matrices mentionnées sont pleines, mais leurs coefficients dépendent de $\mathcal{O}(n)$ paramètres seulement. La notion de rang de déplacement permet d'unifier l'etude de ces structures dans une seule approche. Notons Z la matrice de shift vers le bas, c'est-à-dire

$$Z = \begin{pmatrix} 0 & 0 & \dots & 0 \\ 1 & 0 & \ddots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \dots & 1 & 0 \end{pmatrix}.$$

Le déplacement de la matrice A de taille $n \times n$ vaut $D = A - ZAZ^T$ et le rang de déplacement de A est le rang de déplacement de A est le rang de la matrice D. Un calcul élémentaire montre que si A est une matrice de Toeplitz, les seuls coefficients non nuls de son déplacement sont situés dans sa première ligne et sa première colonne, lesquelles sont respectivement égales à la première ligne et à la première colonne de A. Ainsi D contient toute l'information incluse dans A. Plus généralement, A sera dite de type Toeplitz, si son rang de déplacement est petit devant n. On peut retrouver toute l'information de A à partir de D en remarquant que

$$A = D + ZDZ^T + \cdots + Z^{n-1}D(Z^T)^{n-1}.$$

L'idée de déplacement s'est révélée bien plus puissant est bien plus générale qu'on ne l'avait imaginée. Elle a été étendue aux autres types de matrices structurées et elle joue un rôle central pour construire et analyser les algorithmes [64], [48], [49].

L'étude des matrices structurées nous donne les outils que, logiquement, on doit essayer de les généraliser pour obtenir des algorithmes de résolution rapides de systèmes de Toeplitz par blocs de Toeplitz.

Dans ce chapitre , on va étudier les matrices structurées et les algorithmes associés. Il sera décomposé de la façon suivante :

Dans la section suivante, on rappellera l'algorithme de transformation de Fourier rapide (FFT), et les propriétés des matrices circulantes.

Dans la troisième section, on donnera la définition générale du déplacement pour les structures de Toeplitz, de Hankel, de Vandermonde et de Cauchy. Puis, on donnera les outils de base qui nous permettent de travailler avec les matrices de déplacement. On étudiera ensuite chaque type de matrice structurée. On donnera, enfin les rélations entre les différents types de matrices structurées.

On décrira, dans la section 4, la multiplication rapide d'une matrice structurée par un vecteur.

Dans la dernière section, on discutera les différents types d'algorithmes, rapide et ultra-rapide, de résolution des système structurés.

Dans ce chapitre on ne considère que des matrices carrées, les matrices rectangulaires peuvent être étudiées de la même manière. Si la taille des matrices n'est pas précisée, ce sera $n \times n$, et on utilisera les notations suivantes :

Définition 2.1.1. *J_n (J s'il n'y a pas de confusion) sera la matrice de taille $n \times n$ suivante :*

$$J_n = \begin{pmatrix} & & 1 \\ & \iddots & \\ 1 & & \end{pmatrix}$$

Proposition 2.1.2. *La matrice J est symétrique et orthogonale, donc $J^2 = I$.*

Définition 2.1.3. *On définit Z_φ la matrice suivante :*

$$Z_\varphi = \begin{pmatrix} 0 & 0 & \dots & \varphi \\ 1 & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \dots & 1 & 0 \end{pmatrix},$$

et C_φ l'algèbre engendrée sur $\mathbb{K}$ par Z_φ. La matrice Z_0 sera notée Z, comme on l'a vu précédement.

On peut remarquer facilement que, pour $\varphi \neq 0$, Z_φ est inversible et $Z_\varphi^{-1} = Z_{1/\varphi}^T$.

Définition 2.1.4. *Soit A une matrice par blocs donnée comme suit :*

$$A = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix},$$

telle que A_{11} soit inversible. Le bloc S donné par

$$S = A_{22} - A_{21}A_{11}^{-1}A_{12}$$

s'appelle le complément de Schur de A_{11} dans A.

2.2 Préliminaire

Dans cette section, on rappellera quelques outils qui seront très utile dans la suite de ce travail.

2.2.1 FFT

La transformée de Fourier rapide, FFT (Fast Fourier Transform), est l'un des algorithmes dont la publication a provoqué une véritable révolution dans le calcul. C'est, sans doute, l'algorithme le plus important en mathématiques appliquées et ingénierie. Charles Van Loan a écrit dans son livre [132] : "The fast Fourier transform is one of the truly great computational developments of this century. It has changed the face of science and

engineering so that it is not an exaggeration to say that life as we know it would be very different without FFT". L'importance de la FFT provient de deux facteurs distincts : sa présence dans une pléthore d'applications, dans presque tout les secteurs de la technologie informatique, et la disponibilité d'algorithmes rapides et précis pour la calculer. Il est associé aux noms de James Cooley et John Tuckey qui ont publié cet algorithme en 1965. Il calcule rapidement la transformation de Fourier discrète et il a été redécouvert plusieurs fois depuis Gauss, et surtout par Danielson et Lanczos en 1942. La FFT permet de ramener le calcul de la transformation de Fourier discrète de $\mathcal{O}(n^2)$ flops à $\mathcal{O}(n \log n)$ flops. Voir [62] pour une histoire plus détaillée sur la FFT.

Définition 2.2.1. *La matrice de Fourier d'ordre n est la matrice F_n suivante :*

$$F_n = \frac{1}{\sqrt{n}} \begin{pmatrix} 1 & \dots & \dots & 1 \\ 1 & \omega_n & \dots & \omega_n^{n-1} \\ \vdots & \ddots & \ddots & \vdots \\ 1 & w_n^{n-1} & \dots & \omega_n^{(n-1)^2} \end{pmatrix} = \frac{1}{\sqrt{n}} \left(\omega_n^{ij}\right)_{0 \le i,j \le n-1}.$$

S'il y a pas de confusion on utilise F à la place de F_n.

Définition 2.2.2. *La transformation de Fourier discrète d'un vecteur $v = (v_0, \dots, v_{n-1})^T$ est le vecteur $F\,v$; il sera noté par la suite $\hat{v} = (\hat{v}_0, \dots, \hat{v}_{n-1})^T$.*

Proposition 2.2.3. *La matrice F est symétrique et unitaire.*

Démonstration. La symétrie est évidente. Ecrivons $FF^* = (a_{i,j})_{1 \le i,j \le n}$. Comme $F^H = \frac{1}{\sqrt{n}} \left(\omega_n^{-ij}\right)_{0 \le i,j \le n-1}$, alors

$$a_{i,j} = \frac{1}{n} \sum_{k=1}^{n} \omega_n^{(i-1)(k-1)} \omega_n^{-(k-1)(j-1)} = \frac{1}{n} \sum_{k=1}^{n} \omega_n^{(i-j)(k-1)}.$$

Par suite, si $i = j$ alors $a_{i,i} = \frac{1}{n} \sum_{k=1}^{n} 1 = 1$ et si $i \neq j$ alors $a_{i,j} = 0$ comme somme des exposants d'une racine $n^{\text{ième}}$ de l'unité. □

Remarque 2.2.4. *La matrice de Fourier F_n est la matrice de Vandermonde associée au vecteur $(1, \omega_n, \dots, \omega_n^{n+1})$. Donc, si on associe au vecteur v le polynôme $v(x) = \sum_{i=0}^{n-1} v_i x^i$ alors $\hat{v}$ est simplement le vecteur dont les composantes correspondent à l'évaluation de $v(x)$ aux racines $n^{ième}$ de l'unité.*

Proposition 2.2.5 (Algorithme FFT)**.** *Si n est une puissance de 2, alors on peut calculer $\hat{v}$ en $\frac{3}{2} n \log n$ flops.*

Démonstration. On pose $n = 2m$, et $v^{[0]} = (v_0, v_2, \dots, v_{n-2})^T$, $v^{[1]} = (v_1, v_3, \dots, v_{n-1})^T$; notons $\hat{v}^{[0]}$ et $\hat{v}^{[1]}$ les transformations de Fourier discrètes de taille m de $v^{[0]}$ et $v^{[1]}$ respectivement.

Pour j variant de 0 à $n-1$, on a :

$$\begin{aligned}\hat{v}_j &= \sum_{k=0}^{n-1} v_k \omega_n^{jk} \\ &= \sum_{k=0}^{m-1} v_{2k} \omega_{2m}^{2jk} + \sum_{k=0}^{m-1} v_{2k+1} \omega_{2m}^{2j(k+1)} \\ &= \sum_{k=0}^{m-1} v_{2k} \omega_m^{jk} + \omega_{2m}^{j} \sum_{k=0}^{m-1} v_{2k+1} \omega_m^{jk}\end{aligned}$$

car $\omega_{2m}^{2jk} = e^{2i\pi.2jk/2m} = e^{2i\pi.jk/m} = \omega_m^{jk}$. Donc

$$\begin{cases} \hat{v}_j = \hat{v}_j^{[0]} + \omega_n^j \hat{v}_j^{[1]} & \text{si } \ 0 \leq j \leq m-1 \\ \hat{v}_j = \hat{v}_{j-m}^{[0]} + \omega_n^j \hat{v}_{j-m}^{[1]} & \text{si } \ m \leq j \leq n-1 \end{cases}$$

Ainsi pour calculer la transformation de Fourier d'un vecteur de taille n, il suffit de calculer la transformation de Fourier de deux vecteurs de taille $n/2$ chacun, de faire $n/2$ multiplications et n additions. En effet, pour $m \leq j \leq n-1$, $\hat{v}_j = \hat{v}_{j-m}^{[0]} + \omega_n^j \hat{v}_{j-m}^{[1]} = \hat{v}_j = \hat{v}_{j-m}^{[0]} - \omega_n^{j-m} \hat{v}_{j-m}^{[1]}$, or le nombre $\omega_n^{j-m} \hat{v}_{j-m}^{[1]}$ est déjà calculé, donc pour $m \leq j \leq n-1$ il suffit de faire une addition. Si $c(n)$ est le nombre d'opérations arithmétiques nécessaires pour calculer la transformation de Fourier d'un vecteur de taille n, alors

$$c(n) = 2c(n/2) + 3n/2$$

Par récurrence et en remarquant que $c(1)$ ne demande aucune opération, on a donc

$$c(n) = \frac{3}{2} n \log n.$$

□

2.2.2 Matrice circulante

Les matrices circulantes jouent un rôle important dans l'étude des matrices structurées. C'est pour cela qu'on étudiera ici leurs propriétés.

Définition 2.2.6. *Une matrice $C(r)$ carrée de taille $n \times n$ est dite circulante si elle a la forme suivante :*

$$C(r) = (r \ Z_1 r \ \ldots \ Z_1^{n-1} r)$$

avec $r = (r_0 \ r_1 \ \ldots \ r_{n-1})^T$ la première colonne de C et Z_1 donnée par la définition (2.1.3). Ainsi $C(r)$ est de la forme suivante :

$$C(r) = \begin{pmatrix} r_0 & r_{n-1} & \ldots & r_1 \\ r_1 & r_0 & \ldots & r_2 \\ \vdots & \ddots & \ddots & \vdots \\ r_{n-1} & \ldots & r_1 & r_0 \end{pmatrix}.$$

S'il y a pas de confusion, on utilise C à la place de $C(r)$.

Lemme 2.2.7. *On peut décomposer C de la manière suivante :*

$$C = r_0 I + r_1 Z_1 + \cdots + r_{n-1} Z_1^{n-1}.$$

Démonstration. En utilisant le fait que $Z_1^k e_j = e_{(j+k) \mod n}$ on a

$$\begin{aligned}(r_0 I + r_1 Z_1 + \cdots + r_{n-1} Z_1^{n-1}) e_j &= r_0 + r_1 e_{j+1} + \cdots + r_{n-j} e_n + r_{n-j+1} e_1 \\ &\quad + \cdots + r_{n-1} e_{j-1} = Z_1^{j-1} r = C e_j.\end{aligned}$$

Donc la $j^{\text{ième}}$ colonne de C et de $r_0 I + r_1 Z_1 + \cdots + r_{n-1} Z_1^{n-1}$ sont égales. □

Lemme 2.2.8. *On a $F Z_1 = D F$, où $D = diag(1, \omega, \omega^2, \ldots, \omega^{n-1})$.*

Démonstration. Evident, en comparant les $(i,j)^{\text{ième}}$ coefficients de $F Z_1$ et DF. □

Théoreme 2.2.9. *Si $C(r)$ est une matrice circulante de taille $n \times n$, alors elle est diagonalisable par F. Plus précisement*

$$C = F^* D F,$$

avec $D = diag(Fr)$.

Démonstration.

$$F C F^* = F \left(\sum_{i=0}^{n-1} r_i Z_1^i \right) F^* = \sum_{i=0}^{n-1} r_i F Z_1^i F^* = \sum_{i=0}^{n-1} r_i (F Z_1 F^*)^i = \sum_{i=0}^{n-1} r_i D^i$$

avec $D = \mathrm{diag}(1, \omega, \ldots, \omega^{n-1})$ donc $D^k = \mathrm{diag}(1, \omega^k, \ldots, \omega^{(n-1)k})$ et par suite, en comparant les éléments de la diagonale, on aura $\sum_{i=0}^{n-1} r_i D^i = \mathrm{diag}(Fr)$. □

Corollaire 2.2.10. *Soient $C(r)$ une matrice circulante dont la première colonne est $r = (r_0 \ \ldots \ r_{n-1})^T$ et posont $\hat{r} = (\hat{r}_0 \ \ldots \ \hat{r}_{n-1})^T = Fr$. L'inverse de $C(r)$ est une matrice circulante donnée par $C^{-1}(r) = C(r')$ avec*

$$r' = F^* \left(\frac{1}{\hat{r}_0}, \ldots, \frac{1}{\hat{r}_{n-1}} \right)^T$$

.

Démonstration. On peut décomposer $C(r) = F^* D F$, avec $D = \mathrm{diag}(Fr)$. Donc

$$C^{-1}(r) = F^* D^{-1} F = F^* \mathrm{diag}(\tilde{r}) F = F^* \mathrm{diag}(F F^* \tilde{r}) F = C^{-1}(r'),$$

avec $\tilde{r} = (1/r_0, \ldots, 1/r_{n-1})$. □

Corollaire 2.2.11. *On peut inverser $C(r)$ en $O(n \log n)$ flops.*

Démonstration. Le calcul de $\hat{r}$ coûte une FFT et puis le calcul de r' coûte n flops + une FFT. □

Corollaire 2.2.12. *La multiplication d'une matrice circulante, de taille $n \times n$, par un vecteur coûte $\mathcal{O}(n \log n)$ flops.*

Démonstration. On a $C(r)v = F^* diag(Fr)Fv$, donc la multiplication matrice circulante-vecteur demande trois FFT. □

Corollaire 2.2.13. *L'ensemble des matrices circulantes forme une algèbre commutative.*

Démonstration. Soient C et C' deux matrices circulantes. La matrice $C + C'$ est évidemment une matrice circulante. De plus,

$$CC' = F^*DFF^*D'F = F^*DD'F = F^*D'DF = C'C,$$

avec $D = \mathrm{diag}(Fr)$ et $D' = \mathrm{diag}(Fr')$, où r et r' sont les premières colonnes de C et C' respectivement. donc $CC' = C'C$. De plus elle est circulante de première colonne égale r'' avec $(Fr'')_i = (Fr)_i(Fr')_i$ pour $i = 0, \ldots, n-1$. □

2.2.3 Matrices φ-circulantes

Plus généralement, on peut définir une classe des matrice qui a les mêmes propriétés de la classe des matrices circulantes :

Définition 2.2.14. *Soit $C_\varphi(r)$ une matrice carrée de taille $n \times n$, dont $r = (r_0, \ldots, r_{n-1})^T$ est sa première colonne. $C_\varphi(r)$ est dite φ-circulante si elle a la forme suivante :*

$$C_\varphi(r) = \begin{pmatrix} r_0 & \varphi r_{n-1} & \ldots & \varphi r_1 \\ r_1 & r_0 & \ldots & \varphi r_2 \\ \vdots & \ddots & \ddots & \vdots \\ r_{n-1} & \ldots & r_1 & r_0 \end{pmatrix}.$$

Proposition 2.2.15. *Soit $C_\varphi(r)$ une matrice φ-circulante de taille $n \times n$. elle est aussi diagonalisable par F, et plus précisement on a :*

$$C_\varphi(r) = D_\varphi^{-1} F^* D F D_\varphi,$$

avec $D = diag(FD_\varphi r)$, $D_\varphi = diag(1\, \delta\, \delta^2 \ldots \delta^{n-1})$, pour n'importe quel $\delta \in \mathbb{K}$ qui vérifie $\delta^n = \varphi$.

Démonstration. On peut voir simplement que $Z_\varphi = \delta D_\varphi Z_1 D_\varphi^{-1}$, donc $Z_\varphi^i = \delta^i D_\varphi Z_1^i D_\varphi^{-1}$. Donc on a

$$C_\varphi(r) = \sum_{i=0}^{n-1} r_i Z_\varphi^i = D_\varphi \left(\sum_{i=0}^{n-1} \delta^i r_i Z_1^i \right) D_\varphi^{-1} = D_\varphi C(D_\varphi r) D_\varphi^{-1}$$

□

Proposition 2.2.16. *L'inverse d'une matrice φ-circulante, $C_\varphi(r)$, est φ-circulante de première colonne $F^* \left(1/\hat{r}_0, \ldots, 1/\hat{r}_{n-1}\right)^T$*

Corollaire 2.2.17. *On peut inverser une matrice φ-circulante de taille n en $O(n \log n)$ flops.*

Corollaire 2.2.18. *La multiplication d'une matrice φ-circulante, de taille n et $\varphi \in \mathbb{K}$, par un vecteur coûte $\mathcal{O}(n \log n)$ flops.*

Corollaire 2.2.19. *Pour un $\varphi \in \mathbb{K}$, l'ensemble des matrices $\varphi-$circulantes forme une algèbre commutative.*

Définition 2.2.20. *C_φ est l'algèbre engendrée sur $\mathbb{K}$ par Z_φ.*

Avec cette notation, $\mathcal{C}_0$ est la classe des matrices triangulaires inférieures, $\mathcal{C}_1$ est la classe des matrices circulantes et $\mathcal{C}_\varphi$ est la classe des matrices φ-circulantes.

2.3 Structure de déplacement

Le tableau (2.1) donne les matrices structurées usuelles :

Toeplitz, $T = (t_{i-j})_{i,j=0}^{n-1}$ $T = \begin{pmatrix} t_0 & t_{-1} & \dots & t_{-n+1} \\ t_1 & t_0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & t_{-1} \\ t_{n-1} & \dots & t_1 & t_0 \end{pmatrix}$	**Hankel**, $H = (h_{i+j})_{i,j=0}^{n-1}$ $H = \begin{pmatrix} h_0 & \dots & h_{n-2} & h_{n-1} \\ \vdots & \iddots & \iddots & h_n \\ h_{n-2} & h_{n-1} & \iddots & \vdots \\ h_{n-1} & h_n & \dots & h_{2n-2} \end{pmatrix}$
Vandermonde, $V = V(x) = (x_i^{j-1})_{i,j=1}^{n}$ $V = \begin{pmatrix} 1 & x_1 & \dots & x_1^{n-1} \\ 1 & x_2 & \dots & x_2^{n-1} \\ \vdots & \vdots & & \vdots \\ 1 & x_n & \dots & x_n^{n-1} \end{pmatrix}$	**Cauchy**, $C = C(s,t) = (\frac{1}{s_i - t_j})_{i,j=0}^{n-1}$ $C = \begin{pmatrix} \frac{1}{s_1-t_1} & \dots & \frac{1}{s_1-t_n} \\ \vdots & & \vdots \\ \frac{1}{s_n-t_1} & \dots & \frac{1}{s_n-t_n} \end{pmatrix}$

TAB. 2.1 – Définition des matrices de Toeplitz, de Hankel, de Vandermonde et de Cauchy.

Ces structures et beaucoup d'autres peuvent être unifiées en utilisant le concept de *déplacement*. Ce concept a été introduit en premier temps dans [43] et [70] en relation avec les matrices de Toeplitz. Après les nombreuses applications de cette approche, elle a été étendue significativement en [64] à d'autres modèles de structure.

Rappelons à present des définitions et des observations classiques sur les matrices structurées.

Définition 2.3.1. *Soit A dans $\mathbb{K}^{n\times n}$. Une paire de (G, H), avec G et H dans $\mathbb{K}^{n\times d}$ est appelé un générateur de longueur d de la matrice A si elle peut s'écrire comme produit GH^T. La longueur minimale de d est le rang de A, et sera noté rang(A).*

Les générateurs longs sont simples à calculer, mais les petits générateurs sont les plus intéressants. En particulier, si $r = \text{rang}(A)$ est petit devant n, alors on peut représenter A par $2nr$ coefficients à la place des n^2 coefficients de A. C'est mieux pour le stockage et cela nous permet de multiplier A par un vecteur en $4nr - n - r$ flops à la place de $2n^2 - n$ flops. Supposons que A et B possèdent respectivement un générateur $(G,\, H)$ de longueur r et $(K,\, L)$ de longueur s. Alors (GH^TK, L) est un générateur de AB de longueur s et (G, HK^TL) est un générateur de AB de longueur r. On peut calculer ces générateurs en $4nrs$ flops. Si on veut par exemple les coefficients de AB, la méthode naïve demande $2n^3 + \mathcal{O}(n^2)$ opérations, alors que le passage par les générateurs demande $4nrs + 2n^2 \min(r, s)$ qui est petit devant $2n^3$ si r et s sont petits devant n.

Pour une matrice structurée A, on cherche un opérateur $\mathcal{L}$ qui transforme A en une matrice, $\mathcal{L}(A)$, de petit rang, et telle qu'on puisse facilement retrouver A à partir de son image $\mathcal{L}(A)$.

Définition 2.3.2. *Soit $\mathcal{L} : \mathbb{K}^{n\times n} \longrightarrow \mathbb{K}^{n\times n}$. Soient $A \in \mathbb{K}^{n\times n}$, G, $H \in \mathbb{K}^{n\times d}$ telles que $\mathcal{L}(A) = GH^T$. Alors $r = \text{rang}(\mathcal{L}(A))$ est appelé le $\mathcal{L}$-rang de A et la paire G, H est appelée $\mathcal{L}$-générateur de A de longueur d.*

Le calcul d'un $\mathcal{L}$-générateur de A de longueur minimale se fait par une décomposition LU ou par une décomposition en valeurs singulières (SVD) de $\mathcal{L}(A)$. Un $\mathcal{L}$-générateurs calculé en utilisant la SVD est appelé $\mathcal{L}$-générateur orthogonal. Les générateur orthogonaux sont plus stables pour le calcul numérique, voir [10], [100], [102], [106]. En utilisant une décomposition LU ou une décomposition SVD, le calcul du générateur demandera $\mathcal{O}(n^3)$ flops. Si $\mathcal{L}(A)$ est donnée par un générateur G, H de longueur $R > r$, on peut calculer un générateur de longueur r à partir de G, H en $\mathcal{O}(R^2n)$ flops (voir [9] problème 2.2.11b, [102]).

L'idée des algorithmes adaptés aux matrices structurées repose sur trois principes :

1. **Compression :** la matrice structurée A sera codée par son déplacement $\mathcal{L}(A)$, qui est de petit rang. On a vu, ci dessus, que les opérations matricielles seront plus rapides avec des matrices de rang petit. En pratique $\text{rang}(\mathcal{L}(A))$ doit être très petit devant n pour que ces algorithmes soient efficaces. On a vu aussi, qu'en général, cette étape est coûteuse, parce qu'elle demande $\mathcal{O}(n^3)$ flops. Donc, en général, la matrice A sera donnée par un $\mathcal{L}$-générateur.
2. **Opération :** les opérations sur A peuvent se être traduites en opérations sur $\mathcal{L}(A)$ par des techniques adaptées qui préservent la structure ; elles donnent parfois des générateurs qui ne sont pas minimaux, mais les rendre minimaux ne coûte pas chère, comme on l'a vu ci dessus.
3. **Décompression :** le résultat des opérations se recupére à partir du déplacement.

Définition 2.3.3. *Une matrice A est dite structurée s'il existe un opérateur $\mathcal{L}$ qui satisfait les trois conditions ci-dessus, et telle que $\text{rang}(\mathcal{L}(A))$ est petit devant n.*

2.3.1 Opérateurs de déplacement

On utilisera les deux types d'opérations suivants :

Définition 2.3.4. *L'opérateur de déplacement $D_{M,N}$ est défini comme suit :*

$$\begin{array}{rccc} D_{M,N}: & \mathbb{K}^{n\times n} & \longrightarrow & \mathbb{K}^{n\times n} \\ & A & \longmapsto & MA - AN \end{array} \tag{2.1}$$

Proposition 2.3.5. *L'opérateur $D_{M,N}$ est inversible si et seulement si*

$$spec(M) \cap spec(N) = \emptyset$$

Démonstration. Voir [7]. □

Définition 2.3.6. *On définit l'opérateur $\Delta_{M,N}$ de la manière suivante :*

$$\begin{array}{rccc} \Delta_{M,N}: & \mathbb{K}^{n\times n} & \longrightarrow & \mathbb{K}^{n\times n} \\ & A & \longmapsto & A - MAN \end{array}$$

Pour l'opérateur $\Delta_{M,N}$, on peut pas donner un cas général où cet opérateur sera inversible. Mais on va l'étudier pour chaque type de structure.

Il y a des relations entre les deux types d'opérateurs comme on va le voir à present.

Proposition 2.3.7. *$D_{M,N} = M\Delta_{M^{-1},N}$ si la matrice M est inversible, et $D_{M,N} = -\Delta_{M,N^{-1}}N$ si N est inversible.*

2.3.2 Opérations de base

Notons sur quelques propriétés de base

Proposition 2.3.8. *Soient A, B, M N, $L \in \mathbb{K}^{n\times n}$, alors*

1. $D_{M,N}(A+B) = D_{M,N}(A) + D_{M,N}(B)$,
2. $D_{M,N}(A^T) = -D_{N^T,M^T}(A)^T$,
3. $D_{M,N}(A^{-1}) = -A^{-1}D_{N,M}(A)A^{-1}$,
4. $D_{M,N}(AB) = D_{M,L}(A)B + AD_{L,N}(B)$,
5. $\Delta_{M,N}(A+B) = \Delta_{M,N}(A) + \Delta_{M,N}(B)$,
6. $\Delta_{M,N}(A^T) = \Delta_{N^T,M^T}(A)^T$,
7. $\Delta_{M,N}(AB) = \Delta_{M,L}(A)B + MAD_{L,N}(B)$, *donc*
   (a) $\Delta_{M,N}(AB) = \Delta_{M,L}(A)B + MAL\Delta_{L^{-1},N}(B)$ *si L est inversible,*
   (b) $\Delta_{M,N}(AB) = \Delta_{M,L}(A)B - MA\Delta_{L,N^{-1}}(B)N$ *si N est inversible,*
8. (a) $\Delta_{M,N}(A^{-1}) = MA^{-1}\Delta_{N,M}(A)M^{-1}A^{-1}$ *si M est inversible, et*
   (b) $\Delta_{M,N}(A^{-1}) = A^{-1}N^{-1}\Delta_{N,M}(A)A^{-1}N$ *si N est inversible.*

En particulier, en supposant que $rang_{M,N}(A)$ désigne $rang(D_{M,N}(A))$ ou $rang(\Delta_{M,N}(A))$, on obtient :

1. $rang_{M,N}(A+B) \leq rang_{M,N}(A) + rang_{M,N}(B)$,
2. $rang_{M,N}(A^T) = rang_{N^T,M^T}(A)$,

3. $rang_{M,N}(A^{-1}) = rang_{N,M}(A)$ *(pour* $\Delta_{M,N}$*, il faut que* M *ou* N *soit inversible),*
4. $rang_{M,N}(AB) \leq rang_{M,L}(A) + rang_{L,N}(B)$*. (pour* $\Delta_{M,N}$*, il faut que* M *ou* N *soit inversible).*

Ces formules sont particulièrement sympathiques lorsque $M = N$. Aussi, elles nous permettent de calculer des générateurs pour les matrices produites, en termes des matrices données (sauf pour l'inversion, bien évidemment). En effet,

Proposition 2.3.9. *Soient*

$$\mathbf{D}_{M,N}(A) = G_1 H_1^T, \quad D_{M,N}(B) = G_2 H_2^T \text{ et } D_{N,L}(B) = G_3 H_3^T$$

des générateurs de longueur r_1*,* r_2 *et* r_3 *respectivement, alors*

1. $D_{N^T,M^T}(A^T) = H_1 G_1^T$,
2. $D_{M,N}(A+B) = (G_1|G_2)(H_1|H_2)^T$,
3. $D_{N,M}(A^{-1}) = (-A^{-1}G)(A^{-T}H)^T$,
4. $D_{M,L}(AB) = (G_1|AG_3)(B^T H_1|H_3)^T$.

Ici la notation $(K|L)$ *désigne une matrice formée en concaténant les colonnes de* K *puis celle de* L*.*

Considérons maintenant la décomposition par blocs

$$A = \begin{matrix} \begin{pmatrix} A_{1,1} & A_{1,2} \\ A_{2,1} & A_{2,2} \end{pmatrix} & \begin{matrix} n_1 \\ n_2 \end{matrix} \\ \begin{matrix} n_1 & n_2 \end{matrix} & \end{matrix}$$

et de même pour les matrices M et N. Le résultat suivant fournit des formules pour le déplacement de chacun des blocs. La vérification est directe.

Proposition 2.3.10. *Pour* $1 \leq i, j \leq 2$

$$\mathbf{D}_{M_{i,i},N_{j,j}}(A_{i,j}) = D_{M,N}(A)_{i,j} - A_{i,3-j}N_{3-j,j} + M_{i,3-i}A_{3-i,j}.$$

Maintenant supposons $A_{1,1}$ inversible et soit

$$S = A_{2,2} - A_{2,1}A_{1,1}^{-1}A_{1,2} \in \mathbb{K}^{n_2 \times n_2}$$

le complément de Schur de $A_{1,1}$ dans A. En combinant les trois propositions ci dessus on peut obtenir une expression pour le déplacement de S.

Proposition 2.3.11. *Supposons que* A *et* $A_{1,1}$ *sont inversibles. Si* M *est triangulaire inférieure et* N *est triangulaire supérieure, alors* $rang(D_{M_{2,2},N_{2,2}}(S)) \leq rang(D_{M,N}(A))$*. Si de plus*

$$D_{M,N}(A) = GH^T = \begin{pmatrix} G_1 \\ G_2 \end{pmatrix} \begin{pmatrix} H_1^T & H_2^T \end{pmatrix},$$

alors

$$D_{N_{2,2},M_{2,2}}(S^{-1}) = (B_{2,1}G_1 + B_{2,2}G_2)(B_{1,2}^T H_1 + B_{2,2}^T H_{2,2})^T,$$

avec

$$A^{-1} = \begin{pmatrix} B_{1,1} & B_{1,2} \\ B_{2,1} & B_{2,2} \end{pmatrix}$$

Pour démontrer cette proposition on va utiliser le lemme suivant, qui sera aussi très utile dans la suite :

Lemme 2.3.12. *Soit A une matrice inversible donnée par la décomposition par blocs suivante :*

$$A = \begin{pmatrix} A_{1,1} & A_{1,2} \\ A_{2,1} & A_{2,2} \end{pmatrix} \begin{matrix} n_1 \\ n_2 \end{matrix} \quad ; \\ \begin{matrix} n_1 & n_2 \end{matrix}$$

et si $A_{1,1}$ est inversible, alors :

$$A^{-1} = \begin{pmatrix} A_{1,1}^{-1} + A_{1,1}^{-1}A_{1,2}S^{-1}A_{2,1}A_{1,1}^{-1} & -A_{1,1}^{-1}A_{1,2}S^{-1} \\ S^{-1}A_{2,1}A_{1,1}^{-1} & S^{-1} \end{pmatrix}$$

Démonstration. La décomposition LDU de A est donnée par

$$A = \begin{pmatrix} A_{1,1} & A_{1,2} \\ A_{2,1} & A_{2,2} \end{pmatrix} = \begin{pmatrix} I & 0 \\ A_{2,1}A_{1,1}^{-1} & I \end{pmatrix} \begin{pmatrix} A_{1,1} & 0 \\ 0 & S \end{pmatrix} \begin{pmatrix} I & A_{1,1}^{-1}A_{1,2} \\ 0 & I \end{pmatrix} \tag{2.2}$$

avec $S = A_{2,2} - A_{2,1}A_{1,1}^{-1}A_{1,2}$ le complément de Schur de $A_{1,1}$. Alors

$$\begin{aligned} A^{-1} &= \begin{pmatrix} I & -A_{1,1}^{-1}A_{1,2} \\ 0 & I \end{pmatrix} \begin{pmatrix} A_{1,1}^{-1} & 0 \\ 0 & S^{-1} \end{pmatrix} \begin{pmatrix} I & 0 \\ -A_{2,1}A_{1,1}^{-1} & I \end{pmatrix} \\ &= \begin{pmatrix} A_{1,1}^{-1} + A_{1,1}^{-1}A_{1,2}S^{-1}A_{2,1}A_{1,1}^{-1} & -A_{1,1}^{-1}A_{1,2}S^{-1} \\ S^{-1}A_{2,1}A_{1,1}^{-1} & S^{-1} \end{pmatrix} \end{aligned} \tag{2.3}$$

□

Démonstration. (de la proposition). Comme $D_{M,N}(A) = GH^T$ alors d'après la proposition (2.3.9)

$$\begin{aligned} D_{N,M}(A^{-1}) &= (-A^{-1}G)(A^{-T}H)^T \\ &= \begin{pmatrix} B_{1,1}G_1 + B_{1,2}G_2 \\ B_{2,1}G_1 + B_{2,2}G_2 \end{pmatrix} \begin{pmatrix} H_1^T B_{1,1} + H_2^T B_{2,1} & H_1^T B_{1,2} + H_2^T B_{2,2} \end{pmatrix} \end{aligned}$$

et comme $S^{-1} = (A^{-1})_{2,2}$, alors d'après la proposition (2.3.10)

$$D_{N_{2,2},M_{2,2}}(S^{-1}) = D_{N,M}(A^{-1})_{2,2} = (B_{2,1}G_1 + B_{2,2}G_2)(B_{1,2}^T H_1 + B_{2,2}^T H_2)^T$$

car N est triangulaire supérieure et M est triangulaire. □

Les générateurs ainsi produits ne pas forcément de longueur minimale. Mais comme on a vu, cela ne coûte pas chèr de les rendre minimaux.

2.3.3 Matrice de type Toeplitz

Soit $T = (t_{i-j})_{i,j=0}^{n-1}$ une matrice de Toeplitz de taille n. On remarque facilement que $\forall \varphi, \psi \in \mathbb{K}$ on a :

$$D_{Z_\varphi, Z_\psi}(T) = \begin{pmatrix} \varphi t_{n-1} - t_{-1} & \varphi t_{n-2} - t_{-2} & \dots & \varphi t_1 - t_{-n+1} & \varphi t_0 - \psi t_0 \\ & & & & t_{-n+1} - \psi t_1 \\ & & \bigcirc & & \vdots \\ & & & & t_{-2} - \psi t_{n-2} \\ & & & & t_{-1} - \psi t_{n-1} \end{pmatrix},$$

et que

$$\Delta_{Z_\varphi, Z_\psi^T}(T) = \begin{pmatrix} t_0 - \psi\varphi t_0 & t_{-1} - \varphi t_{n-1} & \dots & t_{-n+1} - \varphi t_1 \\ t_1 - \psi t_{-n+1} & & & \\ \vdots & & \bigcirc & \\ t_{n-1} - \psi t_{-1} & & & \end{pmatrix}.$$

On déduit donc que $\text{rang}(D_{Z_\varphi, Z_\psi}(T)) \leq 2$ et $\text{rang}(\Delta_{Z_\varphi, Z_\psi^T}(T)) \leq 2$. En général, on pose la définition suivante

Définition 2.3.13. *Soit A une matrice de taille $n \times n$. Elle est dite de type Toeplitz si $rang(D_{Z_\varphi, Z_\psi}(T)) \ll n$ ou si $rang(\Delta_{Z_\varphi, Z_\psi^T}(T)) \ll n$.*

On remarque facilement, par exemple, que l'inverse d'une matrice de Toeplitz, n'est pas une matrice de Toeplitz, cependant c'est une matrice de type Toeplitz de rang de déplacement au plus 2. La multiplication de deux matrice de Toeplitz est de type Toeplitz, de rang de déplacement au plus 4, et le complément de Schur dans une matrice de Toeplitz est de type Toeplitz de rang de déplacement au plus 2. Plus généralement on va montrer des résultats reliant une matrice de type Toeplitz aux générateurs de son déplacement par Δ et par D :

Proposition 2.3.14. *Soit A une matrice de type Toeplitz donnée, pour G, $H \in \mathbb{K}^{n \times r}$, et φ, $\psi \in \mathbb{K}^*$ tels que $\varphi\psi \neq 1$, par*

$$\Delta_{Z_\varphi, Z_\psi^T}(A) = GH^T = \sum_{i=1}^{r} g_i h_i^T$$

alors

$$A = \frac{1}{1 - \varphi\psi} \sum_{i=1}^{r} C_\varphi(g_i) C_\psi^T(h_i) \tag{2.4}$$

et, en particulier,

$$\Delta_{Z, Z^T}(A) = GH^T$$

alors

$$A = \sum_{i=1}^{r} L(g_i) L^T(h_i) \tag{2.5}$$

Démonstration. Voir [107] pour la forme (2.4). La forme (2.5) est démontrée par exemple dans [9], théorème 2.11.2. □

Proposition 2.3.15. *Soit A une matrice de type Toeplitz donnée, pour G, $H \in \mathbb{K}^{n\times r}$, et φ, $\psi \in \mathbb{K}^*$ tels que $\varphi \neq \psi$, par*

$$D_{Z_\varphi,Z_\psi}(A) = GH^T = \sum_{i=1}^{r} g_i h_i^T$$

alors

$$A = \frac{1}{\varphi - \psi} \sum_{i=1}^{r} C_\varphi(g_i) C_\psi(Jh_i) \tag{2.6}$$

et si

$$D_{Z,Z}(A) = GH^T$$

alors

$$-A = L(Ae_0) + \sum_{j=1}^{r} L(g_j) L^T(Zh_j) \tag{2.7}$$

Démonstration. Voir [107] et [9], théorème 11.3.2a. □

2.3.4 Matrice de type Hankel

On remarque facilement que les matrices de Hankel et les matrices de Toeplitz sont liées par la proposition suivante :

Proposition 2.3.16. *Si T est une matrice de Toeplitz alors, JT et TJ sont de Hankel, et réciproquement, si H est de Hankel alors, JH et HJ sont de Toeplitz.*

Soit H une matrice de Hankel de taille $n \times n$. Comme dans le cas d'une matrice de Toeplitz, on peut remarquer facilement que $\text{rang}(D_{Z_\varphi,Z_{psi}^T}(H)) \leq 2$ et $\text{rang}(\Delta_{Z_\varphi,Z_{psi}}(H)) \leq 2$. On définit donc les matrice de type Hankel comme suit

Définition 2.3.17. *Soit A une matrice de taille n. La matrice A est dite de type Hankel si $rang(D_{Z_\varphi,Z_\psi^T}(T)) \ll n$ ou $rang(\Delta_{Z_\varphi,Z_\psi}(T)) \ll n$.*

On peut étendre la proposition (2.3.16) pour relier les matrices de type Toeplitz et de type Hankel :

Proposition 2.3.18. *Si A est une matrice de type Toeplitz alors, JA et AJ sont de type Hankel, et vice versa, si B est une matrice de type Hankel alors, JB et BJ sont de type Toeplitz.*

Démonstration. Soit A une matrice de type Toeplitz, c'est-à-dire $\text{rang}(D_{Z_\varphi,Z_\psi})(A) = r$ avec r est petit devant n. On a

$$\begin{aligned} \text{rang}(D_{Z_\varphi,Z_\psi^T})(JA) &= Z_\varphi JA - JAZ_\psi^T = JJZ_\varphi JA - JAZ_\psi^T \\ &= J(JZ_\varphi JA - AZ_\psi^T) = J(Z_\varphi^T A - AZ_\psi^T) \end{aligned}$$

parce que $J^2 = I$ et $JZ_\varphi J = Z_\varphi^T$. La démonstration est terminée car $\text{rang}(J(Z_\varphi^T A - AZ_\psi^T)) = \text{rang}(Z_\varphi A - AZ_\psi)$. □

D'après cette proposition, il suffira d'étudier les matrices de type Toeplitz, puis traduire les conclusions pour les matrices de type Hankel, ce qui revient à inverser l'ordre des inconnues, ou celui des équations.

2.3.5 Matrice de type Vandermonde

Une matrice de Vandermonde associée au vecteur $x = (x_1, \ldots, x_n)$ sera notée $V(x)$ et s'il y a pas de confusion on la notera V. Pour les matrices de Vandermonde on a les déplacements suivants :

Proposition 2.3.19. *Soit $\varphi \in k$ tel que $\varphi x_i^n \neq 1$ pour $i = 1, \ldots, n$,*

$$\Delta_{D(x),Z_\varphi^T}(V) = \begin{pmatrix} 1-\varphi x_1^n & & \\ \vdots & \bigcirc & \\ 1-\varphi x_n^n & & \end{pmatrix} = \begin{pmatrix} 1-\varphi x_1^n \\ \vdots \\ 1-\varphi x_n^n \end{pmatrix} e_1^T,$$

et

$$D_{D^{-1}(x),Z_\varphi^T}(V) = \begin{pmatrix} (1-\varphi x_1^n)/x_1 & & \\ \vdots & \bigcirc & \\ (1-\varphi x_n^n)/x_n & & \end{pmatrix} = \begin{pmatrix} (1-\varphi x_1^n)/x_1 \\ \vdots \\ (1-\varphi x_n^n)/x_n \end{pmatrix} e_1^T.$$

D'après cette proposition, $\text{rang}(\Delta_{D(x),Z_\varphi^T}(V)) = \text{rang}(D_{D^{-1}(x),Z_\varphi^T}(V)) = 1$. On définit donc les matrices de type Vandermonde de la façon suivante :

Définition 2.3.20. *Soit A une matrice de taille $n \times n$. A est dite de type Vandermonde s'il existe un vecteur de longueur n tel que rang$(D_{D^{-1}(x),Z_\varphi^T}(V))$ ou rang$(\Delta_{D(x),Z_\varphi^T}(V))$ est petit devant n.*

Pour reconstruire une matrice de type Vandermonde à partir de son déplacement, on utilise les propositions suivantes :

Proposition 2.3.21. *Soit $\varphi \in k$ tel que $\varphi x_i^n \neq 1$ pour $i = 1, \ldots, n$. Soit A une matrice de type Vandermonde de déplacement*

$$\Delta_{D(x),Z_\varphi^T}(A) = GH^T = \sum_{i=1}^{r} g_i h_i^T,$$

alors

$$A = D(y) \sum_{i=1}^{r} D(g_i) V(x) Z_\varphi^T(h_i)$$

avec $y = (1/(1-\varphi x_i^n))_{i=1}^n$.

Démonstration. Voir [52] □

En utilisant cette proposition et la proposition (2.3.7) on obtient

Proposition 2.3.22. *Soit $\varphi \in k$ tel que $\varphi x_i^n \neq 1$ pour $i = 1, \ldots, n$. Soit A une matrice de type Vandermonde telle que*

$$D_{D^{-1}(x),Z_\varphi^T}(A) = GH^T = \sum_{i=1}^{r} g_i h_i^T,$$

alors

$$A = D(x)D(y)\sum_{i=1}^{r} D(g_i)V(x)Z_\varphi^T(h_i),$$

avec $y = (1/(1-\varphi x_i^n))_{i=1}^n$.

2.3.6 Matrice de type Cauchy

Une matrice de Cauchy associée aux deux vecteurs $s = (s_1, \ldots, s_n)$ et $t = (t_1, \ldots, t_n)$, avec $s_i \neq t_j$ quels que soient i et j, sera notée $C(s,t)$, ou tout simplement C s'il y a pas de confusion. Une matrice de Cauchy $C = C(s,t)$ a le déplacement suivant :

Proposition 2.3.23.

$$\Delta_{D(s)^{-1},D(t)}(C) = \begin{pmatrix} \frac{1}{s_1} \\ \vdots \\ \frac{1}{s_n} \end{pmatrix} . \begin{pmatrix} 1 & \ldots & 1 \end{pmatrix}$$

et

$$D_{D(s),D(t)}(C) = \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} \begin{pmatrix} 1 & \ldots & 1 \end{pmatrix}$$

On remarque que $\text{rang}(\Delta_{D(s)^{-1},D(t)}(C)) = \text{rang}(D_{D(s),D(t)}(C)) = 1$. On définit donc, les matrices de type Cauchy de la façon suivante :

Définition 2.3.24. *Soit A une matrice de taille $n \times n$. A est dite de type Cauchy s'il existe deux vecteurs s et t de longueur n tels que : $rang(D_{D(s),D(t)}(C)) \ll n$.*

On peut reconstruire les matrices de type Cauchy de la manière suivante :

Proposition 2.3.25. *Soit A une matrice de type Cauchy donnée par son déplacement :*

$$\Delta_{D(s)^{-1},D(t)}(A) = GH^T = \sum_{i=1}^{r} g_i h_i^T,$$

alors

$$A = D(s)\sum_{i=1}^{r} D(g_i)C(s,t)D(h_i),$$

et si

$$D_{D(s),D(t)}(A) = GH^T = \sum_{i=1}^{r} g_i h_i^T,$$

alors

$$A = \sum_{i=1}^{r} D(g_i)C(s,t)D(h_i) = \left(\frac{u_i \cdot v_j}{s_i - t_j}\right)_{1\leq i,j\leq n},$$

avec u_i, v_i $1 \leq i \leq n$ *les lignes respectives des matrices* G *et* H.

Démonstration. La démonstration est facile, surtout si on commence par démontrer la deuxième formule ; on utilise ensuite la proposition (2.3.7) pour démontrer la première formule. □

Pour des raisons d'inversibilité des opérateurs, de facilité de reconstruction et de simplicité, on préfère utiliser les opérateurs Δ_{Z,Z^T} ou D_{Z_φ,Z_ψ} pour les matrices de type Toeplitz, $D_{D^{-1}(x),Z^T}$ pour les matrices de type Vandermonde, $D_{D(s),D(t)}$ pour les matrices de type Cauchy.

2.3.7 Relation entre les différents types des matrices structurées

Une relation entre deux types de matrices structurées signifie une transformation rapide du premier type aux deuxième type par multiplication par des matrices qu'on sait les inverser rapidement et les multiplier rapidement par un vecteur. On a déjà vu une relation entre type Toeplitz et type Hankel, puisqu'il suffit de multiplier une matrice de type Toeplitz par la matrice J pour avoir une matrice de type Hankel, et vice versa. Pour trouver des relations entre les autres types, on va utiliser les deux propriétés énoncées à la proposition (2.3.27). Donnons, tout d'abord, quelques définitions.

Définition 2.3.26. *Pour un* $\varphi \in \mathbb{K}^*$, *on note* s_φ *le vecteur de longueur* n

$$s_\varphi = (1, \delta, \ldots, \delta^{n-1}) \text{ pour n'importe quel } \delta \text{ tel que } \delta^n = \varphi.$$

Si $\mathbb{K} = \mathbb{R}$, *les composantes de* s_φ *peuvent être complexe. Pour un vecteur* s *de longueur* n, P_s *désigne la matrice compagnon suivante :*

$$P_s = \begin{pmatrix} 0 & 0 & \ldots & 0 & -p_0 \\ 1 & 0 & \ldots & 0 & -p_1 \\ 0 & 1 & \ldots & 0 & -p_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \ldots & 1 & -p_{n-1} \end{pmatrix}$$

avec $p = (p_0, \ldots, p_{n-1})^T$ *est tel que* $\prod_{i=0}^{n-1}(\lambda - s_i) = \lambda_n + \sum_{i=0}^{n-1} p_i\lambda^i$.

Proposition 2.3.27. *1. Pour* $\varphi \in \mathbb{K}^*$, $Z_\varphi = D(s_\varphi)^{-1}F^*D(\varphi^{1/n}Fe_2)FD(s_\varphi)$.

2. Pour un vecteur $s \in \mathbb{K}^n$, $D(s) = V(s)P_sV(s)^{-1}$ *et* $P_s = Z_\varphi - (p + \varphi e_1)e_n^T$.

Démonstration. Z_φ est une matrice φ circulante. La relation (1) sera donc une conséquence de la décomposition donnée en (2.2.15). Un calcul direct montre que $V(s)P_s = D(s)V(s)$. □

Proposition 2.3.28 (Transformation d'une matrice de type Toeplitz en une matrice de type Cauchy)**.** *Soient $\varphi, \psi \in \mathbb{K}^*$. Soit A une matrice de type Toeplitz donnée par*

$$D_{Z_\varphi, Z_\psi}(A) = GH^T = \sum_{i=1}^{r} g_i h_i^T,$$

alors $\tilde{A} = FD_\varphi A D_\psi^{-1} F^*$ est une matrice de type Cauchy, de rang de déplacement au plus r, donnée par

$$D_{D(s),D(t)}(\tilde{A}) = \tilde{G}\tilde{H}^T = (FD_\varphi G)(F^* D_\psi^{-1} H)^T,$$

avec $s = \varphi^{1/n} F e_2$ et $t = \psi^{1/n} F e_2$.

Démonstration. On a $D_{Z_\varphi, Z_\psi}(A) = Z_\varphi A - A Z_\psi = GH^T$, donc d'après la proposition (2.3.27)

$$D_\varphi^{-1} F^* D(s) F D_\varphi A - A D_\psi^{-1} F^* D(t) F D_\psi = GH^T.$$

On achève la démonstration en factorisant par $D_\varphi^{-1} F^*$ à gauche et par FD_ψ à droite. □

Proposition 2.3.29 (Transformation d'une matrice de type Cauchy en une matrice de type Toeplitz)**.** *Soient s et t deux vecteurs dans $\mathbb{K}^n$. Soit A une matrice de type Cauchy donnée par*

$$D_{D(s),D(t)}(A) = GH^T = \sum_{i=1}^{r} g_i h_i^T,$$

alors $\tilde{A} = V(s)^{-1} A V(t)$ est une matrice de type Toeplitz, de rang de déplacement au plus $r+2$, donnée par

$$D_{Z_\varphi, Z_\psi}(\tilde{A}) = (V(s)^{-1} G | g_{r+1} | g_{r+2})(V(t)^{-1} H | h_{r+1} | h_{r+2})^T,$$

avec $\varphi, \psi \in \mathbb{K}$ et

$$g_{r+1} = p + \varphi e_1,\ g_{r+2} = V(s)^{-1} A V(t)(p' + \psi e_1),$$
$$h_{r+1} = V(t)^{-T} A^T V(s)^{-T} e_n,\ h_{r+2} = e_n.$$

Démonstration. On a $D_{D(s),D(t)}(A) = D(s)A - AD(t) = GH^T$. Or, d'après la proposition (2.3.27),

$$D(s) = V(s) P_s V(s)^{-1} = V(s)(Z_\varphi - (p + \varphi e_1) e_n^T) V(s)^{-1} \text{ et}$$
$$D(t) = V(t) P_t V(t)^{-1} = V(t)(Z_\varphi - (p' + \varphi e_1) e_n^T) V(t)^{-1}$$

donc

$$V(s) Z_\varphi V(s)^{-1} A - A V(t) Z_\psi V(t)^{-1} = GH^T + V(s)(p + \varphi e_1) e_n^T V(s)^{-1} A + A V(t)(p' + \psi e_1) e_n^T(t)$$

On factorise par $V(s)$ à gauche et par $V(t)$ à droite, ce qui nous donne la proposition. □

Proposition 2.3.30 (Transformation d'une matrice de type Toeplitz en une matrice de type Vandermonde)**.** *Soient $\varphi, \psi \in \mathbb{K}^*$. Soit A une matrice de type Toeplitz donnée par*

$$D_{Z_\varphi, Z_\psi}(A) = GH^T = \sum_{i=1}^{r} g_i h_i^T,$$

alors $\tilde{A} = FD_\varphi A$ est une matrice de type Vandermonde donnée par

$$D_{D(s), Z_\psi}(\tilde{A}) = (FD_\varphi G)H = \tilde{G}\tilde{H},$$

avec $s = \varphi^{1/n} Fe_2$.

Démonstration. En suivant la même démarche de la démonstration de la proposition (2.3.28) et en remarquant que $D_{D(s),Z_\psi}$ est une déplacement pour les matrices de type Vandermonde parce que, $D_{D(s),Z_\psi}(A) = D_{D(t)^{-1}, Z^T_{1/\psi}}(AZ^{-1}_{1/\varphi})$, avec $t_i = 1/s_i$. □

Proposition 2.3.31 (Transformation d'une matrice de type Vandermonde en une matrice de type Toeplitz)**.** *Soient $x \in \mathbb{K}^n$, $\varphi \in \mathbb{K}^*$. Soit A une matrice de type Vandermonde donnée par*

$$D_{D_x, Z_\varphi}(A) = GH^T = \sum_{i=1}^{r} g_i h_i^T,$$

alors $\tilde{A} = V(x)^{-1}A$ est une matrice de type Toeplitz, de rang de déplacement au plus $r+1$, donnée par

$$D_{Z_\psi, Z_\varphi}(\tilde{A}) = (V(x)^{-1}G | g_{r+1})(H | h_{r+1})^T,$$

avec $\psi \in \mathbb{K}$, $g_{r+1} = V(x)^{-1}(p + \psi e_1)AV(x)$ et $h_{r+1} = V(x)^{-T} e_n$.

Démonstration. On a $D(x)A - AZ_\varphi = GH^T$. On remplace $D(x)$ par $V(x)(Z_\psi - (p + \psi e_1)e_n)V(x)^{-1}$, puis on factorise par $V(x)$ à gauche. □

Proposition 2.3.32 (Transformation d'une matrice de type Vandermonde en une matrice de type Cauchy)**.** *Soient $x \in \mathbb{K}^n$, $\varphi \in \mathbb{K}^*$. Soit A une matrice de type Vandermonde donnée par*

$$D_{D_x^{-1}, Z_\varphi^T}(A) = GH^T,$$

alors $\tilde{A} = VD_\varphi F$ est une matrice de type Cauchy donnée par

$$D_{D(s), D(t)}(\tilde{A}) = G(FD_\varphi H)^T = \tilde{G}\tilde{H},$$

avec $s = (1/x_1, \ldots, 1/x_n)$ et $t = \varphi^{1/n} Fe_2$.

Démonstration. Même démonstration que pour (2.3.28). □

Proposition 2.3.33 (Transformation d'une matrice de type Cauchy en une matrice de type Vandermonde)**.** *Soient s et t deux vecteurs dans $\mathbb{K}^n$. Soit A une matrice de type Cauchy donnée par*

$$D_{D(s), D(t)}(A) = GH^T = \sum_{i=1}^{r} g_i h_i^T,$$

alors $\tilde{A} = AV(t)$ *est une matrice de type Vandermonde, de rang de déplacement au plus* $r+1$*, donnée par*

$$D_{D(s),Z_\varphi}(\tilde{A}) = (G|g_{r+1})(V(t)^T H|h_{r+1}),$$

avec φ, $\psi \in k$ *et* $g_{r+1} = AV(t)(p + \varphi e_1)$, $h_{r+1} = e_n$.

Démonstration. Même démonstration que pour la proposition (2.3.31). □

On a démontré que la passage entre deux type de matrices structurées se fait au moyen d'une matrice φ circulante ou d'une matrice de Vandermonde et de son inverse. On a déjà vu que la multiplication d'une matrice φ-circulante ou de son inverse par un vecteur ne coûte pas plus que $\mathcal{O}(n \log n)$; on verra aussi que la multiplication d'une matrice de Vandermonde ou de son inverse par un vecteur et le calcul de l'inverse d'une matrice de Vandermonde coûtent $\mathcal{O}(n \log^2 n)$. Par conséquent, on peut passer d'un type à un autre rapidement. Par suite, on peut étendre n'importe quel algorithme de multiplication par un vecteur ou d'inversion d'une matrice structurée de type donné à un autre type de matrice structurée, sans ralentir cet algorithme.

Pour plus d'informations sur la structure de déplacement et sur les matrices structurées, voir les références suivantes : [9], [99], [107], [104], [51], [52], [50], [71], [64], [68].

Dans la section suivante, on va étudier la multiplication d'une matrice structurée par un vecteur. On va voir comment on peut calculer cette multiplication en $\mathcal{O}(rn \log n)$ flops ou $\mathcal{O}(rn \log^2 n)$ flops, où r est le rang de déplacement de la matrice.

2.4 Multiplication rapide

On a vu que la factorisation des matrices structurées contient des matrices de Toeplitz, de Hankel, de Vandermonde et de Cauchy. La possibilité de multiplier rapidement ces quatre types de matrices par un vecteur, conduire à des algorithmes de multiplication rapides pour les matrices structurées.

Proposition 2.4.1. *Soient* T *une matrice de Toeplitz de taille* $n \times n$ *et* v *un vecteur de longueur* n*. On peut calculer le vecteur* Tv *en* $\mathcal{O}(n \log n)$ *flops.*

Démonstration. D'après les propositions (2.3.14) et (2.3.15), une matrice de Toeplitz est la somme de deux matrices qui, elles-mêmes sont produit d'une matrice φ-circulante par une matrice ψ-circulante, avec $\varphi, \psi \in \mathbb{K}^*$. Donc d'après le corollaire (2.2.18) la multiplication Tv ne coûte que $\mathcal{O}(n \log n)$ flops.

Une autre méthode consiste à plonger T dans une matrice circulante de taille $2n$. En effet, si on pose $C = C(r)$ avec

$$r = (t_0, t_1, \ldots, t_{n-1}, t_{-n+1}, t_{-n+2}, \ldots, t_{-1})^T,$$

on a alors

$$C = \left(\begin{array}{c|c} T & T' \\ \hline T' & T \end{array}\right),$$

avec T' une matrice de Toeplitz. Par suite, on peut récupérer Tv en multipliant C par le vecteur, de longueur $2n$, $(v, 0, \ldots, 0)^T$. On a l'idendité

$$C \begin{pmatrix} v \\ 0 \end{pmatrix} = \begin{pmatrix} Tv \\ T'v \end{pmatrix}.$$

La multiplication d'une matrice de Toeplitz par un vecteur coûte donc $\mathcal{O}(n \log n)$.

On peut voir enfin la multiplication d'une matrice de Toeplitz par un vecteur comme une multiplication de deux polynômes. Notons $T(x) = \sum_{i=-n+1}^{n-1} t_i x^{i+n-1}$ et $w(x) = T(x)v(x)$. L'écriture matricielle de cette multiplication est

$$\begin{pmatrix} w_0 \\ \vdots \\ \\ \\ \vdots \\ w_{3n-3} \end{pmatrix} = \begin{pmatrix} t_{-n+1} & & & 0 \\ \vdots & \ddots & & \\ t_0 & \ddots & \ddots & t_{-n+1} \\ \vdots & \ddots & \ddots & \vdots \\ t_{n-1} & \ddots & \ddots & t_0 \\ & \ddots & \ddots & \vdots \\ 0 & & & t_{n-1} \end{pmatrix} \begin{pmatrix} v_0 \\ v_1 \\ \vdots \\ v_{n-1} \end{pmatrix}, \tag{2.8}$$

donc le vecteur Tv égale $(w_{n-1}, \ldots, w_{2n-1})^T$. Or la multiplication de deux polynômes de degré n et m respectivement est de l'ordre $(m+n)\log(m+n)$ flops, voir [9] problème 1.2.4a, ce qui donne une troisième estimation et une troisième méthode. □

Une conséquence de cette proposition est que la multiplication d'une matrice de Hankel par un vecteur se fait en $\mathcal{O}(n \log n)$ flops.

Soient $t = (t_1, \ldots, t_n) \in \mathbb{K}^n$, $V = V(t)$ la matrice de Vandermonde associée au vecteur t, $v = (v_0, v_1, \ldots, v_{n-1})$ un vecteur de longueur n et w le vecteur $w = Vv$. Alors w_i n'est jamais que l'évaluation du polynôme $p(x) = \sum_{i=0}^{n-1} v_i x^i$ au point t_i pour $i = 1, \ldots, n$. Or d'après le problème 1.2.2 de [9], l'évaluation d'un polynôme de degré $n-1$ sur n points demande $\mathcal{O}(n \log^2 n)$ flops. On peut déduire donc la proposition suivante :

Proposition 2.4.2. *Soient V une matrice de Vandermonde, v un vecteur. Le calcul de Vv peut se faire en $\mathcal{O}(n \log^2 n)$ flops.*

Pour traiter le cas de Cauchy, nous avonsbesoin du calcul préparatoire suivant :

Soient s et t deux vecteurs de $\mathbb{K}^n$. Soient $C = C(s,t)$ la matrice de Cauchy associée aux vecteurs s et t, $v = (v_1, \ldots, v_n) \in \mathbb{K}^n$ et $w = Cv$; on a donc pour $1 \leq i \leq n$ la relation

$$w_i = \sum_{j=1}^{n} \frac{v_j}{s_i - t_j}.$$

Or le polynôme d'interpolation de degré $n-1$ qui vérifie $p(x_i) = y_i$, pour $(x_i, y_i) \in \mathbb{K}^2$ et $i = 1, \ldots, n$, est donné par

$$p(x) = L(x) \sum_{j=1}^{n} \frac{y_j}{(x - x_j)L'(x_j)},$$

avec $L(x) = \prod_{i=1}^{n}(x - x_i)$. Dans notre cas nous remplaçons les x_i par les t_i et nous avons la formule suivante

$$w_i = \sum_{j=1}^{n} \frac{v_j}{s_i - t_j} = \frac{1}{L(s_i)} L(s_i) \sum_{j=1}^{n} \frac{L'(t_j)v_j}{(s_i - t_j)L'(t_j)} = \frac{p(s_i)}{L(s_i)},$$

$p(x)$ étant le polynôme de degré $n-1$ donné par

$$p(t_i) = L'(t_i)v_i \text{ pour } i = 1, \ldots, n.$$

On peut maintenant démontrer la propriété suivante :

Proposition 2.4.3. *Soient $C = C(s,t)$ une matrice de Cauchy associée aux vecteurs s et t, et v un vecteur dans $\mathbb{K}^n$. On peut calculer $w = Cv$ en $\mathcal{O}(n \log^2 n)$.*

Démonstration. Comme on l'a vu, pour calculer $w = Cv$, il faut évaluer $L(x)$ sur les n points $s_1, \ldots, s_n$ et le polynôme $L'(x)$ sur $t_1, \ldots, t_n$, puis calculer le polynôme d'interpolation qui vérifie $p(s_i) = L'(t_i)v_i$ pour $i = 1, \ldots, n$. L'évaluation des deux polynômes coûte $\mathcal{O}(n \log^2 n)$ et l'interpolation coûte aussi $\mathcal{O}(n \log^2 n)$, voir [9] problème 1.2.3. □

En utilisant ces trois propositions on peut donner une méthode de multiplication rapide d'une matrice structurée par un vecteur. Plus généralement, on peut décrire comment multiplier deux matrices structurées, de rang de déplacement r et r' respectivement, en moins de $\mathcal{O}(rr'n \log^2 n)$ flops.

Définition 2.4.4. *On note $mv_r(\mathcal{L})$, le nombre d'opérations arithmétiques nécessaires pour multiplier une matrice structurée par un vecteur, donnée par son $\mathcal{L}-$générateur de longueur r. On note $m_{r,r}(\mathcal{L}, \mathcal{L}')$, le nombre d'opérations nécessaires pour multiplier deux matrices structurées données par leurs $\mathcal{L}$ et $\mathcal{L}'-$générateurs de longueur r et r' respectivement.*

Définition 2.4.5. *Les paires d'opérateurs : $(D_{M,L}, D_{L,N})$, $(\Delta_{M,L}, D_{L,N})$, $(\Delta_{M,L}, \Delta_{L^{-1},N})$, et $(\Delta_{M,L}, \Delta_{L,N^{-1}})$ sont dits compatibles.*

Proposition 2.4.6. *On a :*

1. *$mv_r(\mathcal{L}) = \mathcal{O}(rn \log n)$ pour $\mathcal{L} = D_{M,N}$ ou $\mathcal{L} = \Delta_{M,N}$, et M, $N \in \{Z_\varphi, Z_\varphi^T\}$ quel que soit φ,*
2. *$mv_r(\mathcal{L}) = \mathcal{O}(rn \log^2 n)$ pour $\mathcal{L} = D_{M,N}$ ou $\mathcal{L} = \Delta_{M,N}$, avec $M = D(s)$ et $N = D(t)$, ou $M = D(s)$ et $N \in \{Z_\varphi, Z_\varphi^T\}$, ou $M \in \{Z_\varphi, Z_\varphi^T\}$ et $N = D(s)$, quels que soient les vecteurs s et t et quel que soit φ dans $\mathbb{K}$.*
3. *pour une paire d'opérateurs compatibles $\mathcal{L}$ et $\mathcal{L}'$, on a : $m_{r,r'}(\mathcal{L}, \mathcal{L}') = \mathcal{O}(r'mv_r(\mathcal{L}) + rmv'_r(\mathcal{L}'))$.*

Démonstration. Si M et N sont données comme dans 1), alors la matrice sera de type Toeplitz ou de type Hankel. Donc une telle matrice peut s'écrire comme somme, de longueur r, d'une matrice φ-circulante par une matrice ψ-circulante, voir propositions (2.3.14) et (2.3.15). Comme la multiplication d'une matrice φ circulante par un vecteur coûte

$\mathcal{O}(n \log n)$ flops, alors la multiplication d'une matrice de type Toeplitz, de rang de déplacement r, par un vecteur coûte $\mathcal{O}(rn \log n)$ flops. Il en est de même dans le deuxième cas, mais la matrice sera de type Vandermonde ou de type Cauchy, et la multiplication d'un vecteur par une matrice de Cauchy ou de Vandermonde coûte $\mathcal{O}(n \log^2 n)$.

La troisième propriété dérive de la première et deuxième propriété de cette proposition et des propriétés, d'une matrice produit de deux matrices structurées, donnée en (2.3.8) et (2.3.9) □

On se reportera à [107], [52], [9], pour plus d'information sur la multiplication d'une matrice structurée par un vecteur.

2.5 Algorithmes rapides et ultra-rapides de résolution des systèmes à matrice structurée

Un algorithme rapide de résolution est un algorithme qui demande $\mathcal{O}(n^2)$ flops pour résoudre un système linéaire de taille $n \times n$, et algorithme ultra-rapide un algorithme qui demande un nombre d'opérations petit devant n^2.

Avant de commencer à étudier les algorithmes de résolution des systèmes à matrice structurée en général, on commencera par les cas particuliers des matrices de Toeplitz de Vandermonde et de Cauchy.

2.5.1 Résolution d'un système de Toeplitz

On peut définir une matrice de Toeplitz entièrement à partir de sa première et sa dernière colonne. On peut étendre cette propriété à l'inverse d'une matrice de Toeplitz grâce à la formule de Gohberg-Semencul. Cette propriété dérive du fait que l'inverse d'une matrice de Toeplitz est aussi de rang de déplacement au plus 2 ainsi que de la nature des opérateurs de déplacement pour une matrice de Toeplitz. Commençons par démontrer la formule de Gohberg-Semencul. Soit T une matrice de Toeplitz de taille $n \times n$, qu'on va supposer inversible. On a

$$D_{Z,Z}(T) = ZT - TZ = -e_1 \hat{u}^T + u e_n^T \tag{2.9}$$

avec $u = ZTe_n$. Pour un vecteur v, on note, $\hat{v} = Jv$. En multipliant (2.9) par T^{-1} à gauche et à droite, on obtient

$$D_{Z,Z}(T^{-1}) = T^{-1} e_1 \hat{u}^T T^{-1} - T^{-1} u e_n^T T^{-1}$$

Soit x, z tels que $Tx = e_1$ et $Tz = u$; en utilisant la symétrie de T par rapport à l'antidiagonale et l'identité $e_n = \hat{e}_0$ on déduit que

$$D_{Z,Z}(T^{-1}) = x\hat{z}^T - z\hat{x}^T$$

En utilisant la proposition (2.3.15) on obtient

$$T^{-1} = -L(x) - L(x)L^T(Z\hat{z}) + L(z)L^T(Z\hat{x}) = L(z)L^T(Z\hat{x}) - L(x)(I_n + L^T(Z\hat{z})) \quad (2.10)$$

$$= \begin{pmatrix} z_1 & & & \\ z_2 & z_1 & & \\ \vdots & \ddots & \ddots & \\ z_n & \dots & z_2 & z_1 \end{pmatrix} \begin{pmatrix} 0 & x_n & \dots & x_2 \\ & \ddots & \ddots & \vdots \\ & & \ddots & x_n \\ & & & 0 \end{pmatrix} - \begin{pmatrix} x_1 & & & \\ x_2 & x_1 & & \\ \vdots & \ddots & \ddots & \\ x_n & \dots & x_2 & x_1 \end{pmatrix} \begin{pmatrix} 1 & z_n & \dots & z_2 \\ & \ddots & \ddots & \vdots \\ & & \ddots & z_n \\ & & & 1 \end{pmatrix}$$

Remarque 2.5.1. *En utilisant cette formule d'inversion on peut donner une formule de récurrence pour calculer les coefficients de T^{-1}. En effet, en écrivant $T^{-1} = (c_{ij})_{i,j=1}^n$ et d'après* (2.10) *on aura*

$$c_{i,j} = \sum_{k=0}^{\min(i,j)} z_{i-k}x_{n-j+k} - x_{i-k}z_{n-j+k}$$

donc on a la formule de récurrence :

$$c_{i,j} = c_{i-1,j-1} + z_i x_{n-j} - x_i z_{n-j}, \quad i,j = 2,\dots,n$$

Maintenant, en prenant y tel que $Ty = e_n$ on aura :

Lemme 2.5.2. *On peut écire z en fonction de x et y de la façon suivante :*

$$z = \frac{1}{x_1}((\hat{u}^T.y)x - Zy)$$

Démonstration. D'après (2.9) on a

$$D_{Z,Z}(T)y = TZy - ZTy = e_1\hat{u}^T.y - ue_n^T.y$$

comme $ZTy = Ze_n = 0$ et $Tx = e_1$ et comme $y_1 = x_1$ on obtient

$$TZy = Tx\hat{u}^T y - uy_1 = (\hat{u}^T.y)Tx - x_1 u$$

par suite

$$T((\hat{u}^T.y)x - Zy) = x_1 u$$

□

En remplaçant z par sa valeur dans l'équation (2.10), on obtient la formule de Gohberg-Semencul suivante :

Corollaire 2.5.3. *Si $x_1 \neq 0$ alors T^{-1} est donnée par*

$$T^{-1} = \frac{1}{x_1}\left(\begin{pmatrix} x_1 & & & \\ x_2 & x_1 & & \\ \vdots & \ddots & \ddots & \\ x_n & \dots & x_2 & x_1 \end{pmatrix} \begin{pmatrix} x_1 & y_n & \dots & y_2 \\ & \ddots & \ddots & \vdots \\ & & \ddots & y_n \\ & & & x_1 \end{pmatrix} - \begin{pmatrix} 0 & & & \\ y_1 & \ddots & & \\ \vdots & \ddots & \ddots & \\ y_{n-1} & \dots & y_1 & 0 \end{pmatrix} \begin{pmatrix} 0 & x_n & \dots & x_2 \\ & \ddots & \ddots & \vdots \\ & & \ddots & x_n \\ & & & 0 \end{pmatrix}\right),$$

c'est-à-dire

$$x_1 T^{-1} = L(x)L^T(x_1 e_1 + Z\hat{y}) - L(Zy)L^T(Z\hat{x}),$$

avec x la première colonne de T^{-1} et y est sa dernière colonne.

Remarque 2.5.4. *Grâce à la règle de Cramer, $x_1 \neq 0$ si et seulement si T_{n-1} est inversible.*

L'importance de cette formule est que si on peut décomposer l'inverse d'une matrice de Toeplitz T selon cette formule, alors la résolution de n'importe quel système $Tx = b$ coûtera seulement $\mathcal{O}(n \log n)$ pour chaque b donné.

Plusieurs formes de cette formule sont données dans la litérature. On peut trouver dans [64] une autre forme qui donne T^{-1} en fonction de sa première et sa deuxième colonnes et en supposant $x_n \neq 0$. Dans [78], les auteurs démontrent que T^{-1} peut être reconstruite à partir de n'importe quelles deux colonnes de T^{-1} et quelques coefficients de la matrice T. Une forme circulante de cette formule est donnée dans [2] et [51]. On donnera, dans le chapitre (6), une nouvelle forme de la formule de Gohberg-Semencul presentée d'un autre point de vue, en relation avec les syzygies. Pour plus d'information sur ce sujet, on peut voir [64], [78], [65], [2], [51], [67], [26], [92]

Les systèmes linéaires de Toeplitz sont très bien étudié dans la littérature et ils sont le sujet d'un grand nombre de publications qui exploitent leur structure pour donner des algorithmes rapides de résolution.

Il y a deux types d'algorithmes directes rapids, qui demandent $\mathcal{O}(n^2)$ flops pour résoudre un système de Toeplitz : algorithme de type Levinson (qu'on va le détailler dans la suite) et de type Schur. En principe, les algorithmes de type Levinson donnent une décomposition LU de l'inverse de la matrice, et les algorithmes de type Schur donnent une décomposition LU de la matrice elle même, voir par exemple, [64], [106], [45], [24], [61], [89] pour les algorithmes de type Levinson et [71], [106], [118], [3], [45], [61] pour les algorithmes de type Schur. Ces deux types d'algorithmes peuvent être étendus naturellement aux matrices de type Toeplitz, avec une complexité de $\mathcal{O}(rn^2)$ flops pour une matrice de taille $n \times n$ et de rang de déplacement r. Voir par exemple [89], [64] pour les algorithmes de type Levinson et [118] pour les algorithmes de type Schur.

Ces deux types d'algorithmes ne marchent que dans le cas où les sous-matrices principales de la matrice T sont inversibles. Numériquement, cela signifie une instabilité si les sous-matrices principales sont mal conditionnées. Une technique pour résoudre ce problème d'instabilité est de regarder d'avance le conditionement à chaque étape et de sauter les étapes instables. On se référera par exemple à [23], [24], [61], [128], [127]. Une autre idée pour résoudre ce problème d'instabilité est de transformer la matrice de Toeplitz en une matrice de type Cauchy ou de type Vandermonde.

Les algorithmes de résolution de complexité petit devant $\mathcal{O}(n^2)$ sont appelés des algorithmes ultra-rapides. Les solveurs ultra-rapides sont basés sur l'idée de diviser pour régner. Le principe d'un solveur ultra-rapide de systèmes de Toeplitz est donné pour la première fois dans [85]. Plus tard, d'autres algorithmes basés sur la même idée sont donnés dans [11], [16]. Dans [86], l'auteur donne un algorithme ultra-rapide pour les matrices de type Toeplitz. Dans [104], l'auteur donne un algorithme ultra-rapide, basé sur la même

idée que dans les références précédentes mais qui étendue à n'importe quelle matrice structurée. L'idée de ces algorithmes est de décomposer la matrice, notée A et de taille $n \times n$, en blocs de taille $n/2$, puis d'utiliser les formules de décomposition et d'inversion données en (2.2) et (2.3). Pour calculer ces deux formules, on réapplique l'algorithme à $A_{1,1}$ et S et à $A_{1,1}^{-1}$ et S^{-1}. Ainsi de suite, jusqu'à se réduire à des matrices de taille 1×1. Pour une matrice structurée, on calcule récursivement les générateurs en profitant du fait que les sous-matrices, les compléments de Schur et l'inverse des compléments de Schur héritent de la structure de la matrice. Les solveurs super rapides peuvent arriver à une complexité en $\mathcal{O}(r^2 n \log^2 n)$ pour une matrice structurée de taille $n \times n$ et de rang de déplacement r.

Des algorithmes de Schur super rapides, de complexité $\mathcal{O}(n \log^2 n)$ ou $\mathcal{O}(n \log^3 n)$ pour des algorithmes plus stables, sont donnés par exemple dans [4], [5], [118], [89], [33], [3] pour des matrices de Toeplitz. D'autres types d'algorithmes super rapides sont donnés dans [129], [97] et dans le chapitre (6) de cette thèse.

Il faut bien remarquer que tous ces algorithmes, utilisent le fait qu'une matrice de Toeplitz est de rang de déplacement au plus 2, pour obtenir cette complexité. Cette remarque est importante car le rang de déplacement d'une matrice de Toeplitz par blocs de Teoplitz est généralement grand devant 2.

Des algorithmes itératifs sont aussi donnés dans la littérature. Les algorithmes itératifs reposent sur la multiplication rapide d'une matrice de Toeplitz par un vecteur. Dans [22], les auteurs utilisent une méthode de gradient conjugué préconditionné pour des matrices de Toeplitz symétriques définies positives. Ainsi, ils donnent un algorithme de résolution ultra-rapide. Dans leur article, ils proposent plusieurs formes de préconditionneurs. Un algorithme qui utilise les itérations de Newton, est proposé dans [107], [106]. La référence [73] utilise un algorithme basé sur l'algorithme de Weidemann. Ce dernier est une méthode d'espaces de Krylov. L'algorithme de [73] coûte $\mathcal{O}(n^2 \log n)$ flops, mais il est bien adapté au calcul parallèle. Cet algorithme, qui est utilisé principalement pour les matrices creuses, peut être généralisé facilement au cas de matrices de Toeplitz par blocs de Toeplitz, parce qu'il utilise seulement, comme dans le cas des matrices creuses, la multiplication rapide, ici celle des matrices de Toeplitz ou de Toeplitz par blocs de Toeplitz par un vecteur.

Algorithme de Levinson

L'algorithme de Levinson est un algorithme de résolution d'un système de Toeplitz, qui construit récursivement la solution, ce qui va nous donner une résolution en $\mathcal{O}(n^2)$ flops. Cet algorithme dans le chapitre 3, dans le cas d'une matrice de Toeplitz par blocs et de Toeplitz par blocs de Toeplitz, et il est presenté ci-après.

Soit le système $Tx = b$, avec $b = (b_1, \ldots, b_n)^T$, $x = (x_1, \ldots, x_n)^T$ et T comme dans le tableau (2.1), page 20.

A chaque étape on cherche X_{k+1} à partir de X_k, où X_k est la solution du système $T_k X_k = B_k$, avec $B_k = (b_1, \ldots, b_k)^T$ et T_k la sous-matrice principale de T de taille $k \times k$.

On partitionne la matrice T_{k+1} par blocs $k \times k$, $1 \times k$, $k \times 1$ et 1×1 et on écrit la

décomposition LU pa rblocs sous la forme suivante.

$$T_{k+1} = \left(\begin{array}{c|c} T_k & \hat{V}_k \\ \hline \hat{W}_k^T & t_0 \end{array}\right) = \begin{pmatrix} I_k & 0 \\ \hat{W}_k^T T_k^{-1} & 1 \end{pmatrix} \begin{pmatrix} T_k & \hat{V}_k \\ 0 & s_k \end{pmatrix} \tag{2.11}$$

Ici, $V_k = (t_{-1}, \ldots, t_{-k})^T$, $W_k = (t_1, \ldots, t_k)^T$, $Y_k^T = V_k^T T_k^{-1}$ et $Z_k = T_k^{-1} W_k$, et pour un vecteur v, $\hat{v} = Jv$. De plus, $s_k = t_0 - W_k^T J T_k^{-1} J V_k = t_0 - W_k^T T^{-T} V_k = t_0 - W_k^T (V_k^T T^{-1})^T = t_0 - (Y_k W_k)^T$.

On écrit

$$X_{k+1} = \begin{pmatrix} \nu_k \\ x_{k+1} \end{pmatrix}.$$

Miltiplions (2.11) à gauche par l'inverse du facteur triangulaire inférieur, et appliquons le résultat à $B_{k+1} = T_{k+1} X_{k+1}$. Il vient :

$$\begin{pmatrix} T_k & \hat{V}_k \\ 0 & s_k \end{pmatrix} \begin{pmatrix} \nu_k \\ x_{k+1} \end{pmatrix} = \begin{pmatrix} I_k & 0 \\ -\hat{W}_k^T T_k^{-1} & 1 \end{pmatrix} \begin{pmatrix} B_k \\ b_{k+1} \end{pmatrix},$$

ce qui équivaut au système

$$\begin{cases} s_k x_{k+1} = -\hat{W}_k^T X_k + b_{k+1} \\ T_k \nu_k + \hat{V}_k x_{k+1} = B_k. \end{cases}$$

De là, on tire les valeurs de x_{k+1} et ν_k

$$\begin{cases} x_{k+1} = \dfrac{-\hat{W}_k^T X_k + b_{k+1}}{s_k} \\ \nu_k = X_k - T^{-1} J V_k x_{k+1} \\ \quad = X_k - J(V_k^T T_k^{-1})^T x_{k+1} \\ \quad = X_k - x_{k+1} \hat{Y}_k \end{cases},$$

et la nouvelle expression de X_{k+1}

$$X_{k+1} = \begin{pmatrix} X_k - x_{k+1} \hat{Y}_k \\ x_{k+1} \end{pmatrix}, \tag{2.12}$$

avec

$$x_{k+1} = \frac{-\hat{W}_k^T U_k + b_{k+1}}{s_k}, \text{ et } X_1 = x_1 = \frac{b_1}{t_0}. \tag{2.13}$$

On a donc une manière pour calculer X_{k+1} à partir de X_k. Il reste à calculer Y_k, Z_k et s_k. On pose

$$Y_{k+1}^T = \begin{pmatrix} \mu_k \\ y_{k+1} \end{pmatrix} = V^T T_{k+1}^{-1}.$$

En multipliant (2.11) à gauche par $(\mu_k^T \; y_{k+1})$, il vient

$$(\mu_k^T \; y_{k+1}) \begin{pmatrix} I_k & 0 \\ \hat{W}_k^T T_k^{-1} & 1 \end{pmatrix} = (V_k^T \; t_{-k-1}) \begin{pmatrix} T_k^{-1} & -T_k^{-1} \hat{V}_k s_k^{-1} \\ 0 & s_k^{-1} \end{pmatrix},$$

ce qui donne par le même raisonnement que pour X_{k+1}

$$Y_{k+1}^T = \begin{pmatrix} Y_k^T - y_{k+1}\hat{Z}_k^T \\ y_{k+1} \end{pmatrix}, \tag{2.14}$$

avec

$$y_{k+1} = -\frac{Y_k^T\hat{V}_k + t_{-k-1}}{s_k}, \text{ et } Y_1 = y_1 = \frac{t_{-1}}{t_0}. \tag{2.15}$$

Pour calculer des Z_k, on pose

$$Z_{k+1} = \begin{pmatrix} \beta_k \\ z_{k+1} \end{pmatrix}.$$

Un calcul similaire nous donne

$$Z_{k+1} = \begin{pmatrix} Z_k - z_{k+1}\hat{Y}_k \\ z_{k+1} \end{pmatrix}, \tag{2.16}$$

avec

$$z_{k+1} = \frac{-\hat{W}_k^T Z_k + t_{k+1}}{s_k}, \text{ et } Z_1 = \frac{t_1}{t_0}. \tag{2.17}$$

De plus on peut calculer s_{k+1} récursivement à partir de s_k de la façon suivante :

$$\begin{aligned} s_{k+1} &= t_0 - Z_{k+1}^T V_{k+1} = t_0 - (Z_k^T - z_{k+1}\hat{Y}_k^T \; z_{k+1}) \begin{pmatrix} V_k \\ t_{-k-1} \end{pmatrix} \\ &= t_0 - Z_k^T V_k + z_{k+1}\hat{Y}_k^T V_k + z_{k+1}t_{-k-1} = s_k + z_{k+1}y_{k+1}s_k \\ &= (1 + z_{k+1}y_{k+1})s_k. \end{aligned}$$

A chaque étape, $k = 1, \ldots, n$, cet algorithme demande $k+1$ flops pour calculer chacun des x_k, y_k et z_k, et il demande $2k-2$ flops pour calculer chacun de ν_k, μ_k et β_k, et 3 flops pour calculer s_k. Ceci fournit un algorithme en $\mathcal{O}(n^2)$ flops.

Algorithme de Schur

L'idée de cet algorithme, pour une matrice de Toeplitz symétrique définie positive, est d'utiliser le déplacement de T donné par

$$\Delta_{Z,Z^T}(T) = T - ZTZ^T = G\Sigma G^T, \tag{2.18}$$

$$G^T = \frac{1}{\sqrt{t_0}} \begin{pmatrix} t_0 & t_1 & \ldots & t_{n-1} \\ 0 & t_1 & \ldots & t_{n-1} \end{pmatrix},$$

et

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

On remarque que $G\Sigma G^T = (GA)\Sigma(GA)^T$ pour n'importe quelle matrice A qui vérifie $A\Sigma A^T = \Sigma$. Toute matrice qui vérifie cette propriété sera notée Σ-unitaire. Toutes les matrices Σ-unitaires sont de la forme suivante

$$A = \frac{1}{1-|\rho|} \begin{pmatrix} a & 0 \\ 0 & b \end{pmatrix} \begin{pmatrix} 1 & \bar{\rho} \\ \rho & 1 \end{pmatrix}$$

avec a, b et ρ sont dans $\mathbb{K}$ et $|a| = |b| = 1$. C'est-à-dire que les matrices Σ-unitaires est le produit d'une matrice de rotation hyperbolique par une matrice diagonale unitaire.

On choisit donc A comme produit d'une matrice de rotation hyperbolique par une matrice diagonale unitaire de façon que $\tilde{G} = GA$ soit de la forme

$$\tilde{G} = \begin{pmatrix} \tilde{g}_{11} & 0 \\ \tilde{g}_{21} & \tilde{g}_{22} \\ \vdots & \vdots \\ \tilde{g}_{n1} & \tilde{g}_{n2} \end{pmatrix}.$$

On peut remarquer simplement que la première ligne de T égale $\tilde{g}_{11}(\tilde{g}_{11} \ldots \tilde{g}_{n1}$. Ainsi, la première ligne de $\tilde{G}^T$ est la première ligne de la décomposition de Cholesky de T. En répétant la même chose sur le complément de Schur de t_0, qui vérifie aussi une déplacement semblable à celle de l'équation (2.18), on obtiendra la deuxième ligne de la décomposition de Cholesky. On continue de la même manière jusqu'on obtient toute la décomposition de Cholesky de T.

Cet algorithme demande $\mathcal{O}(n^2)$ flops pour achever la décomposition de T. Une vesion ultra-rapide de cet algorithme est possible en donnant une relation entre les matrices A et les polynômes. Voir [118].

2.5.2 Résolution d'un système de Vandermonde

Soit $V = V_x$ une matrice de Vandermonde, de taille $n \times n$, associée au vecteur x. La solution du système $Vp = b$ est le vecteur des coefficients du polynôme d'interpolation, $p(x)$, de degré $n-1$, qui vérifie $p(x_i) = b_i$ pour $i = 1, \ldots, n$. Donc la résolution d'un système de Vandermonde peut se faire en $\mathcal{O}(n \log^2 n)$ d'après [9].

De plus pour obtenir la matrice V^{-1} il suffit de résoudre seulement deux systèmes linéaires ; en effet, on a vu que

$$D_{D(x),Z^T}(V) = D^{-1}(x)V - VZ^T = we_1^T, \tag{2.19}$$

avec $w_i = 1/x_i$ pour $i = 1, \ldots, n$. Soient u, v tels que $Vu = w$ et $V^T v = e_1$, c'est-à-dire $u = V^{-1}w$ et $v^T = e_1^T V^{-1}$. En multipliant (2.19) par V^{-1} à gauche et à droite on obtient

$$V^{-1}D^{-1}(x) - Z^T V^{-1} = -D_{D^{-1}(x),Z^T}(V^{-1}) = uv^T.$$

D'après la proposition (2.3.21) on obtient la formule suivante d'inversion d'une matrice de Vandermonde :

$$V^{-1} = -D(x)D(u)V(x)L(v).$$

2.5.3 Résolution d'un système de Cauchy

Soit $C = C_{s,t}$ une matrice de Cauchy ; on s'interesse au système $Cx = b$.

On a déjà vu en étudiant la multiplication d'une matrice de Cauchy par un vecteur une identité intéressante. En effet, si on pose $L(x) = \prod_{i=1}^{n}(x - t_i)$, alors

$$\forall i = 1, \ldots, n,\ b_i = \frac{p(s_i)}{L(s_i)} \text{ avec } p(t_i) = L'(t_i)x_i.$$

Donc, pour $i = 1, \ldots, n$,

$$x_i = \frac{p(t_i)}{L'(t_i)},$$

où $p(x)$ est le polynôme d'interpolation, de degré $n-1$, donné par

$$p(s_i) = L(s_i)b_i \text{ pour } i = 1, \ldots, n.$$

Ensuite, pour résoudre $Cx = b$, il faut résoudre un problème d'interpolation de degré $n-1$, puis évaluer deux polynômes de degré $n-1$ en n points, ce qui donne un algorithme qui coûte $\mathcal{O}(n \log^2 n)$ flops.

Comme dans le cas d'une matrice de Toeplitz ou de Vandermonde, et en utilisant la structure de déplacement de la matrice C, on peut réconstruire son inverse à partir de deux vecteurs. En effet,

$$D_{D(s),D(t)}(C) = D(s)C - CD(t) = \mathbf{1}\mathbf{1}^T, \tag{2.20}$$

avec $\mathbf{1} = (1, \ldots, 1)^T$. Soient x, y tels que $Cx = \mathbf{1}$ et $C^T y = \mathbf{1}$ c'est-à-dire $x = C^{-1}\mathbf{1}$, $y^T = \mathbf{1}^T C^{-1}$.

En multipliant (2.20) par C^{-1} à gauche et à droite, on obtient

$$C^{-1}D(s) - D(t)C^{-1} = -D_{D(t),D(s)}(C^{-1}) = C^{-1}\mathbf{1}\mathbf{1}^T C^{-1} = xy^T.$$

Ainsi, on a la formule suivante d'inversion d'une matrice de Cauchy :

$$C^{-1} = -\text{diag}(x)C_{s,t}\text{diag}(y) = -\left(\frac{x_i y_j}{t_i - s_j}\right).$$

Le calcul de x coûte $\mathcal{O}(n \log^2 n)$ flops. Le calcul de y coûte, en utilisant la même technique, $\mathcal{O}(n \log^2 n)$ flops parce que $C^T = C(t, s)$ est aussi une matrice de Cauchy, ce qui donne un total de $\mathcal{O}(n \log^2 n)$ flops.

2.5.4 Résolution d'un système à matrice structurée

Une transformation rapide entre les différents types de matrices structurées est possible. On peut ainsi étendre un algorithme de résolution, rapide ou ultra-rapide, d'un système à matrice structurée de type donné à un autre type.

Une généralistion stricte des algorithmes de Levinson et de Schur peut donner des algorithmes, rapides ou ultra-rapides, de résolution des matrices de type Toeplitz. Voir [89], [64], [118].

La matrice de type Cauchy est invariante par permutation de lignes ou de colonnes. Soit C une matrice de type Cauchy de rang de déplacement r, c'est-à-dire qu'il existe n vecteurs u_i et n vecteurs v_i de longueur r tels que

$$C = \left(\frac{u_i.v_j}{s_i - t_j}\right)_{1 \le i,j \le n}.$$

On remarque facilement que la permutation de deux lignes de C ne change pas sa forme générale, elle restra donc une matrice de type Cauchy de rang de déplacement r. On peut

profiter de cette proriété pour donner, facilement, des algrithmes de Gauss avec pivot adaptés aux matrices de type Cauchy, voir [50] et [95].

La simplicité de sa structure est une autre propriété trés importante d'une matrice de type Cauchy. La structure d'une matrice de type Cauchy est définie par deux matrices diagonales, ce qui donne donc la structure la plus simple parmi les matrices structurées. En profitant de cette simplicité de structure on peut donner des algorithmes ultra-rapides d'inversion d'une matrice de type Cauchy, voir [108].

Un algorithme ultra-rapide adapté à n'importe quel type de matrices structurées est donné dans [104].

Chapitre 3

Où trouve-t-on des matrices de Toeplitz par blocs de Toeplitz

3.1 Introduction

Dans le premier chapitre on a étudié les matrices structurées scalaires. Mais les matrices structurées à deux niveaux et en particulier les matrices de Toeplitz par blocs de Toeplitz (TBT) apparaissent dans beaucoup d'applications. En général, les matrices obtenues sont de taille très grande, et donc difficiles à manipuler. L'élimination de Gauss standard, qui demande $\mathcal{O}(n^3)$ flops n'est pas acceptable dans des tels cas. Même des méthodes de complexité $\mathcal{O}(n^2)$ restent lentes. On voit donc que des algorithmes de résolution rapide pour des systèmes structurés à deux niveaux ou plus seront cruciaux. Le but de cette thèse était de trouver des algorithmes de résolution de complexité petite devant n^2.

Avant de s'attaquer à ce problème, on va donner dans ce chapitre quelques applications où on trouve des matrices de Toeplitz par blocs de Toeplitz.

Dans les équations aux dérivées partielles, dès qu'on a un problème non trivial, nous trouverons des matrices de Toeplitz par blocs de Toeplitz, en dimension 2, pour un maillage uniforme et des coefficients constants ; en dimension 3, la généralisation est évidente : matrice de Toeplitz à trois niveaux. Si on tient compte des conditions aux limites dans des cas suffisamment généraux, on devra passer de matrices TBT à des objets plus généraux, par exemple les matrices de type TBT.

En géométrie algébrique, les relations entre polynômes en plusieurs variables et matrices structurées multidimentionelles sont évidentes. Multiplier deux polynômes en deux variables est équivalent à multiplier une matrice TBT par un vecteur, [88]. Les matrices, de résultant, de Sylvester, de Bézout, de Macaulay, utilisées en théorie de l'élimination sont des matrices structurées multiniveaux, [39]. La résolution des systèmes polynomiaux spéciaux peut se réduire à des systèmes linéaires dont la matrice est Toeplitz multiniveaux, [87], [13]. On va voir dans le chapitre 6 une relation entre la résolution d'un système à matrice TBT et les syzygies des polynômes en deux variables.

Les systèmes TBT paraissent dans plusieurs problèmes de traitement d'images. L'absence de formule d'interpolation de Lagrange en deux dimensions (2D) impose de passer par un problème de moindres carrés en deux dimensions, ce qui donne une matrice TBT.

L'équation de Yule-Walker pour une prédiction linéaire 2D dans une corps aléatoire homogène en deux dimensions a une matrice TBT. La déconvolution par moindres carrés en 2D d'une image à partir d'une fonction d'étalement de point donne une matrice TBT. Il y a bien sûr encore beaucoup d'autres applications.

On va décrire plus précisément l'obtention de matrices TBT dans les équations aux dérivées partielles(EDP), en géométrie algébrique et en traitement d'images et du signal dans les deuxième, troisième et quatrième sections respectivement.

3.2 EDP et matrices TBT

On commence par l'exemple du laplacien sur un carré. On va discrétiser cette équation par un schéma aux différences finies uniformes et avec des conditions de Dirichlet au bord. On cherche $u(x, y)$ telle que

$$-\Delta u = f \text{ dans le carré, } u = 0 \text{ sur le bord,} \tag{3.1}$$

avec f une fonction donnée. Pour discrétiser cette équation, on va supposer que la longueur du côté du carré vaut 1 et on va le discrétiser uniformement en prenant des point de discrétisation (x_i, y_j) pour $1 \leq i, j \leq m$. Le pas de discrétisation est $\Delta x = 1/(m+1)$ et l'équation (3.1) se discrétise donc en

$$\frac{U_{i,j+1} + U_{i,j-1} + U_{i-1,j} + U_{i+1,j} - 4U_{i,j}}{\Delta x^2} = F_{i,j}, \tag{3.2}$$

avec $F_{i,j} = f(x_i, y_j)$ et $U_{i,j}$ les valeurs approchées de $u(x_i, y_j)$ qu'on cherche. On doit bien sûr étendre la définition de $U_{i,j}$ aux bords du carré discret. On le prend nul pour $i = 0$ ou n, ou $j = 0$ ou n.

Dans ce cas, si on ordonne les points du carré dans l'ordre

$$(1,1), \ldots, (n,1), (1,2), \ldots, (n,2), \ldots, (1,n), \ldots (n,n) \tag{3.3}$$

la matrice du système (3.2) est donnée par

$$A = \begin{pmatrix} a & b & 0 & \ldots & 0 \\ b & a & b & \ldots & 0 \\ 0 & b & a & \ldots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & \ldots & a \end{pmatrix},$$

$$\text{avec } a = \frac{1}{\Delta x^2} \begin{pmatrix} 4 & -1 & 0 & \ldots & 0 \\ -1 & 4 & -1 & \ldots & 0 \\ 0 & -1 & 4 & \ldots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & \ldots & 4 \end{pmatrix} \text{ et } b = \frac{1}{\Delta x^2} \begin{pmatrix} 0 & -1 & 0 & \ldots & 0 \\ -1 & 0 & -1 & \ldots & 0 \\ 0 & -1 & 0 & \ldots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & \ldots & 0 \end{pmatrix}.$$

On a ainsi un exemple tout à fait classique de matrice TBT, qui est, ici, en plus bande par blocs bande.

La structure de Toeplitz paraît aussi en passant aux éléments finis. Rappelons quelques définitions et résultats classiques [110] ou [112]

Définition 3.2.1. *On définit $H^1(0,1)$ comme le sous-espace de $L^2(0,1)$ formé des fonctions dont la dérivée au sens des distributions est aussi dans $L^2(0,1)$.*

Définition 3.2.2. *L'espace $H_0^1(0,1)$ est le sous-espace des fonctions u de $H^1(0,1)$ nulles au bord.*

Théoreme 3.2.3. *Muni de la norme*

$$\left(\int_0^1 (|u|^2 + |u'|^2)\, dx\right)^{1/2},$$

$H^1(0,1)$ est un espace de Hilbert dont tous les éléments sont presque partout égaux à une fonction continue sur $[0,1]$.

Dans le cas de dimension 1, considérons le problème aux limites suivants

$$-(au')' = f, \text{ sur }]0,1[\text{ et } u(0) = u(1) = 0, \tag{3.4}$$

avec $a \in C^0(0,1)$ et $0 < \alpha \leq a$ presque partout sur $[0,1]$, pour une certaine constante strictement positive α. Nous supposons que f est dans $L^2(0,1)$. Ce problème possède une unique solution, comme on le voit à l'aide de la technique suivante de noyau.

S'il existe une fonction u de classe C^1, telle que u' soit dans H_0^1, qui résout le problème (3.4), alors on peut intégrer l'équation (3.4) sur $[0,x]$:

$$-a(x)u'(x) + a(0)u'(0) = \int_0^x f(y)dy$$

et donc, en posant $p = u'(0)$ et en intégrant encore une fois par rapport à x :

$$u(x) = p\int_0^x \frac{a(0)}{a(y)}dy - \int_0^x f(z)\int_z^x \frac{dy}{a(y)}\, dz.$$

On peut trouver la valeur de p en utilisant la condition $u(1) = 0$. On obtiendra finalement la formule suivante

$$u(x) = \int_0^1 K(x,y)f(y)dy$$

avec

$$K(x,y) = \begin{cases} \dfrac{b(y)(b(1)-b(x))}{b(1)} & \text{si } 0 \leq y \leq x \\ \dfrac{b(x)(b(1)-b(y))}{b(1)} & \text{si } x \leq y \leq 1 \end{cases}$$

et $b(x) = \int_0^1 dy/a(y)$.

En particulier, au' est dans $H^1(0,1)$ et si a est dans $H^1(0,1)$, alors u' lui-même est dans $H^1(0,1)$. Si f est continue, alors u est de classe C^2.

La formulation faible ou variationnelle du problème repose sur la remarque suivante : si v appartient à $C^1(0,1)$ et est nul en 0 et 1, alors on peut multiplier par v l'équation

$-(au')' = f$, intégrer sur $[0,1]$ le résultat et procéder à une intégration par parties du terme comportant des dérivées, afin d'obtenir la relation suivante :

$$\int_0^1 au'v'dx = \int_0^1 fv\,dx. \tag{3.5}$$

Il est commode de poser $\mathcal{A}(u,v) = \int_0^1 au'v'dx$ et $\mathcal{L}(v) = \int_0^1 fv\,dx$. Réciproquement, si (3.5) est vérifiée quel que soit v dans un espace fonctionnel convenable, alors on récupère (3.4). Ce résultat classique est décrit par la proposition suivante :

Proposition 3.2.4. *Notons X l'ensemble des fonctions de classe C^1 sur $[0,1]$ nulles aux extrémités et $V = H_0^1(0,1)$. Pour tout $f \in L^2(0,1)$ (resp. $f \in C^0(0,1)$) et tout $a \in L^\infty(0,1)$ (resp. $a \in C^0(0,1)$) et a minoré par une constante $\alpha > 0$, il existe un unique $u \in V$ (resp. $u \in X$) tel que pour tout $v \in V$ (resp. $v \in X$) on ait*

$$\mathcal{A}(u,v) = \mathcal{L}(v).$$

De plus, au' est dans $H^1(0,1)$ (resp. $C^0(0,1)$) et on a , au sens de distributions,

$$-(au')' = f.$$

Démonstration. Voir par exemple [110] ou [112]. □

On peut montrer aussi que u est l'unique solution du problème d'optimisation sur $H_0^1(0,1)$ de la fonctionnelle suivante :

$$J(u) = \frac{1}{2}\mathcal{A}(u,u) - \mathcal{L}(v). \tag{3.6}$$

L'intérêt pour l'analyse numérique de la formulation faible est d'introduire une notion de solution prise dans un espace nettement plus grand que celui qu'imposerait le résultat final de régularité. Elle permet également de formuler des approximations en dimension finie, en remplaçant l'espace V par un espace d'approximation de dimension finie, et si nécessaire, $\mathcal{A}$ et $\mathcal{L}$ par des approximations.

Plus précisement, si V_h et W_h sont des sous-espaces de dimension finie de V, on peut s'intéresser au problème suivant : trouver $u_h \in V_h$ tel que pour tout $v \in W_h$ on ait

$$\mathcal{A}(u_k,v) = \mathcal{L}(f,v). \tag{3.7}$$

Il faut tout d'abord trouver des conditions nécessaires et suffisantes pour que (3.7) possède une solution. Si $V_h = W_h$, la restriction de $\mathcal{A}$ à $V_h \times V_h$ est une forme bilinéaire symétrique définie positive. Il est donc clair que (3.7) possède une unique solution.

Le cas des éléments finis est obtenu en choisissant des points $x_0 = 0 < x_1 \cdots < x_{n-1} < 1 = x_n$ dans l'intervalle $[0,1]$ et en prenant comme V_h l'ensemble des fonctions continues sur $[0,1]$, nulles aux extrémités, et dont la restriction à chaque intervalle $]x_j, x_{j+1}[$ est un polynôme de degré au plus d. Ainsi V_h est l'espace des éléments finis $\mathbb{P}_d$.

Pour améliorer l'efficacité de la méthode, il convient de remplacer les intégrales exactes par des formules d'intégration numérique. Ainsi, dans le cas des éléments finis $\mathbb{P}_1$, on choisit la formule des trapèzes, c'est-à-dire

$$\int_{x_j}^{x_{j+1}} g(x)dx \backsim \frac{g(x_j)+g(x_{j+1})}{2}(x_{j+1}-x_j).$$

Cela revient à remplacer respectivement $\mathcal{A}$ et $\mathcal{L}$ par

$$\mathcal{A}_h(u,v) = \sum_{j=1}^{n-1} a(x_j)u'(x_j)v'(x_j)\frac{x_{j+1}-x_{j-1}}{2}$$

et

$$\mathcal{L}_h(v) = \sum_{j=0}^{n-1} f(x_j)v(x_j)\frac{x_{j+1}-x_{j-1}}{2}.$$

Afin de résoudre effectivement, on choisit une base de V_h. On choisit la base φ_i donnés par

$$\varphi_i(x) = \begin{cases} \dfrac{x-x_{i-1}}{x_i-x_{i-1}} & \text{sur } [x_{i-1},x_i] \\ \dfrac{x_{i+1}-x}{x_{i+1}-x_i} & \text{sur } [x_i,x_{i+1}] \\ 0 & \text{ailleurs} \end{cases}$$

forment une base de V.

Alors l'équation (3.7) est vérifiée pour tout $v \in V$ si et seulement si elle est vérifiée pour tout les φ_i. Donc u est une solution de (3.4) si et seulement si

$$\int_0^1 a\left(\sum_{i=1}^{n} u(x_i)\varphi_i'(x)\right)\varphi_j'(x)\,dx = \int_0^1 f\varphi_j(x)\,dx.$$

Les inconnues sont les $u_i = u(x_i)$ pour $0 \le i \le n$. La matrice A du système a pour coefficients

$$A_{ij} = \int_0^1 a(x)\varphi_i'(x)\varphi_j'(x)\,dx.$$

Comme l'intersection du support de φ_i avec le support de φ_j est d'intérieur non vide seulement si $|i-j| \le 1$, les A_{ij} sont nuls si $|i-j| > 1$. La matrice des A_{ij} est donnée par

$$A = (A_{ij})_{i,j} = \begin{pmatrix} \gamma_2+\gamma_1 & -\gamma_2 & & & & \\ -\gamma_2 & \gamma_3+\gamma_2 & -\gamma_3 & & & \\ \ddots & \ddots & \ddots & & & \\ & \ddots & \ddots & \ddots & & \\ & & -\gamma_{n-1} & \gamma_n+\gamma_{n-1} & -\gamma_n & \\ & & & -\gamma_n & \gamma_{n+1}+\gamma_n \end{pmatrix}$$

avec

$$\gamma_i = \frac{a(x_i) - a(x_{i-1})}{2(x_i - x_{i-1})} \text{ pour } 1 \leq i \leq n+1.$$

A est symétrique, définie positive et tridiagonale. si $x_{i+1} - x_i$ ne dépend pas de i (discrétisation uniforme) et si a est une constante strictement positive, alors A est de Toeplitz.

Passons au cas des éléments $\mathbb{P}_2$. Une base de l'espace V_h est donnée par

$$\varphi_j(x) = \begin{cases} \dfrac{4(x_{j+1} - x)(x - x_j)}{(x_{j+1} - x_j)^2} & \text{si } x \in [x_j, x_{j+1}] \\ 0 & \text{ailleurs} \end{cases} \qquad 0 \leq j \leq n-1$$

$$\psi_j(x) = \begin{cases} \dfrac{(x_{j+1} - x)(x_{j+1} + x_j - 2x)}{(x_{j+1} - x_j)^2} & \text{si } x \in [x_j, x_{j+1}] \\ \dfrac{(x - x_{j-1})(2x - x_j - x_{j-1})}{(x_j - x_{j-1})^2} & \text{si } x \in [x_{j-1}, x_j] \\ 0 & \text{ailleurs} \end{cases} \qquad 0 \leq j \leq n-1$$

La forme de la matrice A dépend bien sûr de l'ordre de ces fonctions de base. Choisissons l'ordre suivant :

$$\varphi_0, \psi_1, \varphi_1\psi_2, \ldots, \varphi_{n-2}, \psi_{n-1}, \varphi_{n-1}.$$

Les propriétés de support impliquent :

$$\begin{aligned} \mathcal{A}_h(\varphi_i, \varphi_j) &= 0 \quad \text{si} \quad i \neq j \\ \mathcal{A}_h(\varphi_i, \psi_j) &= 0 \quad \text{si} \quad j \neq i, i+1 \\ \mathcal{A}_h(\psi_i, \psi_j) &= 0 \quad \text{si} \quad |i - j| > 1. \end{aligned}$$

La matrice A sera donc de la forme :

$$\begin{pmatrix} p_{0,0} & p'_{0,1} & & & & & & \\ p'_{0,1} & q_{1,1} & p'_{1,1} & q'_{1,2} & & & & \\ & p'_{1,1} & p_{1,1} & p'_{1,2} & & & & \\ & q'_{1,2} & p'_{1,2} & q_{2,2} & p'_{2,2} & q'_{2,3} & & \\ & & & p'_{2,2} & p_{2,2} & p'_{2,3} & & \\ & & & q'_{2,3} & p'_{2,3} & q_{3,3} & p'_{3,3} & q'_{3,4} & \\ & & & & \ddots & \ddots & \ddots & \ddots & \ddots \end{pmatrix}$$

avec $p_{i,i} = \mathcal{A}_h(\varphi_i, \varphi_i)$, $p'_{i,j} = \mathcal{A}_h(\varphi_i, \psi_j)$, $q_{i,i} = \mathcal{A}_h(\psi_i, \psi_i)$, $q'_{i,j} = \mathcal{A}_h(\psi_i, \psi_j)$. Dans l'hypothèse d'un a constant et d'une subdivision uniforme, et en posant

$$\begin{aligned} p &= \mathcal{A}_h(\varphi_0, \varphi_0) \qquad & p' &= \mathcal{A}_h(\varphi_0, \psi_1) \\ q &= \mathcal{A}_h(\psi_1, \psi_1) \qquad & q' &= \mathcal{A}_h(\psi_1, \psi_2) \end{aligned}$$

la matrice devient

$$\begin{pmatrix} p & p' & & & & & & \\ p' & q & p' & q' & & & & \\ & p' & p & p' & & & & \\ & q' & p' & q & p' & q' & & \\ & & & p' & p & p' & & \\ & & & q' & p' & q & p' & q' \\ & & & & \ddots & \ddots & \ddots & \ddots & \ddots \end{pmatrix}.$$

Ici, on a utilisé la symétrie $\mathcal{A}_h(\varphi_0, \psi_1) = \mathcal{A}_h(\psi_1, \varphi_1)$, et la structure par blocs est claire.

C'est dans le cas bidimensionnel que l'apparition de structures de Toeplitz biniveau ou multiniveau est frappante.

Limitons-nous encore une fois à un problème elliptique donné sous la forme variationnelle : trouver $u \in V$ tel que pour tout $v \in V$ on a

$$\mathcal{A}(u, v) = \mathcal{L}(v)$$

et à une méthode d'éléments finis. A cette fin, on se fixe un maillage du domaine ω où est posée l'équation aux dérivées partielles, on prend comme espace d'approximation V_h un espace de fonctions continues dont les restrictions à chaque élément du maillage est dans un espace approprié de polynômes. Cette présentation permet de prendre en compte également des systèmes d'équations elliptiques et donc des espaces de polynômes plus sophistiqués que l'espace des polynômes de degré total au plus k (éléments finis $\mathbb{P}_k$) ou de degré partiel au plus k (éléments finis $\mathbb{Q}_k$). Voir par exemple [90] ou [40] ou encore [27] pour des descriptions approfondies d'espaces d'éléments finis.

Le raisonnement qui suit repose uniquement sur les propriétés des supports des éléments de la base de l'espace V_h.

Supposons d'abord notre système d'équations aux dérivées partielles à coefficients constants. Si τ est une translation arbitraire dans le plan, notons

$$u^\tau(x) = u(\tau x).$$

Alors, quelles que soient les fonctions u et v on aura

$$\mathcal{A}(u^\tau, v^\tau) = \mathcal{A}(u, v)$$

pourvu que les supports de u, v, u^τ, v^τ soient entièrement inclus dans ω.

Supposons maintenant que le maillage utilisé soit l'intersection de ω avec un maillage pavant régulièrement le plan.. Notons $(x_i)_{1 \le i \le N}$ un ensemble de noeuds associés à la méthode et au maillage, et φ_i la base de V_h définie par

$$\varphi_i(x_j) = \delta_{ij}.$$

On peut choisir les noeuds de telle façon que l'ensemble des indices des φ_i puisse être partitionné en sous ensembles $P_1, \ldots, P_r$ de façon que tous les φ_i pour $i \in P_j$ se déduisent les uns des autres par translation.

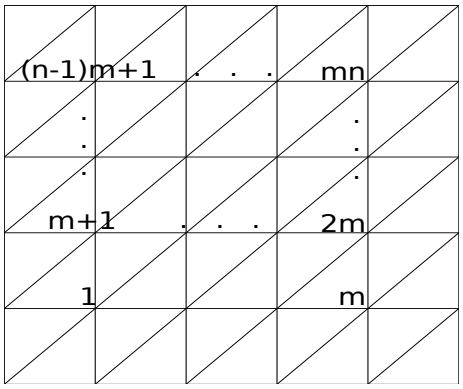


FIG. 3.1 – Maillage par des triangles rectangles à mn noeuds.

D'après les propriétés du maillage, on pourra donc numéroter les x_i de façon que P_j soit une réunion disjointe d'ensemble de la forme

$$\{m_{jp} + k_{jp}l,\ 1 \leq l \leq l_{jp}\},\ 1 \leq p \leq P_j$$

et en particulier si ω est un rectangle on aura

$$P_j = \{m_j + k_j l + q_j r,\ 1 \leq l \leq l_j,\ 1 \leq r \leq r_j\}.$$

Il est alors à peu près évident que A aura une structure de Toeplitz bande multiniveau, mais étant donnée la complexité d'une formulation très générale, on se contentera de traiter quelques exemples significatifs.

Exemple 1

Discrétisation $\mathbb{P}_1$ avec maillage par des triangles rectangles uniforme, voir figure. (3.1) : Dans ce cas, tous les φ_i se déduisent les uns des autres par translation, avec deux translation de base :

$$\begin{aligned} i &\longmapsto i+1 \quad \text{si} \quad i \notin m\mathbb{N} \\ i &\longmapsto i+m \quad \text{si} \quad i \leq (m-1)m \end{aligned}$$

Les coefficients A_{ij} sont donc donnés par :

$$\mathcal{A}(\varphi_i, \varphi_j) = \begin{cases} \mathcal{A}(\varphi_1, \varphi_1) & \text{si} \quad j = i \\ \mathcal{A}(\varphi_1, \varphi_2) & \text{si} \quad j = i+1 \\ \mathcal{A}(\varphi_1, \varphi_{m+1}) & \text{si} \quad j = i+m,\ i \leq (n-1)m \\ \mathcal{A}(\varphi_1, \varphi_{m+2}) & \text{si} \quad j = i+m+1,\ i \leq (n-1)m,\ i \notin m\mathbb{N} \\ 0 & \text{partout ailleurs} \end{cases}.$$

Ce résultat se traduit en numérotation à double indice définie par

$$i = mq + r,\ 0 \leq r < m$$

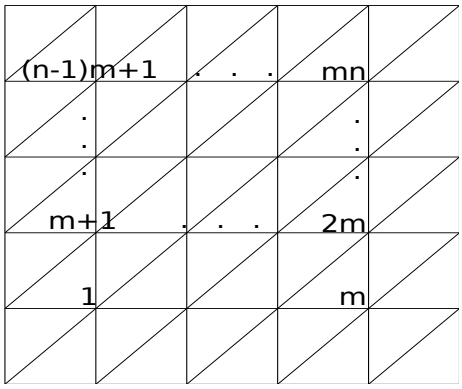


FIG. 3.2 – Maillage par des triangles rectangles avec sous-structure.

sous la forme

$$Aqr, q'r' = b_{|q-q'|,|r-r'|}$$

avec $b_{0,0} = \mathcal{A}(\varphi_1, \varphi_1)$, $b_{0,1} = \mathcal{A}(\varphi_1, \varphi_2)$, $b_{1,0} = \mathcal{A}(\varphi_1, \varphi_{m+1})$, $b_{1,1} = \mathcal{A}(\varphi_1, \varphi_{m+2})$, $b_{ij} = 0$ pour tous autres choix de i et j. Ce qui fournit bien une structure de Toeplitz bande par blocs de Toeplitz bande.

Exemple 2

Discrétisation $_{|1}$ avec maillage par des triangles rectangles avec sous structure, voir figure (3.2) : Cette structure a évidement un petit air académique, mais elle a l'avantage par rapport à la première structure de ne pas oublier aux coins sud-ouest et nord-est tout un demi-triangle. Pour qu'il en soit ainsi, il faut que m et n soient impairs tous les deux.

Posons :

$$\begin{array}{ccc} \mathcal{A}(\varphi_1, \varphi_1) = \alpha, & \mathcal{A}(\varphi_2, \varphi_2) = \beta, & \mathcal{A}(\varphi_1, \varphi_2) = \gamma, \\ \mathcal{A}(\varphi_1, \varphi_{n+1}) = \delta, & \mathcal{A}(\varphi_1, \varphi_{n+2}) = \epsilon, & \mathcal{A}(\varphi_3, \varphi_{n+3}) = \zeta. \end{array}$$

posons aussi :

$$M = \begin{pmatrix} \beta & \gamma & & & & & \\ \gamma & \alpha & \gamma & & & & \\ & \gamma & \beta & \gamma & & & \\ & & \gamma & \alpha & \gamma & & \\ & & & \gamma & \beta & \gamma & \\ & & & & \ddots & \ddots & \ddots \\ & & & & & \gamma & \beta \end{pmatrix}, \quad L = \begin{pmatrix} \alpha & \gamma & & & & & \\ \gamma & \beta & \gamma & & & & \\ & \gamma & \alpha & \gamma & & & \\ & & \gamma & \beta & \gamma & & \\ & & & \gamma & \alpha & \gamma & \\ & & & & \ddots & \ddots & \ddots \\ & & & & & \gamma & \alpha \end{pmatrix},$$

$$P = \begin{pmatrix} \delta & \epsilon & & & & & \\ & \delta & & & & & \\ & \zeta & \delta & \epsilon & & & \\ & & & \delta & & & \\ & & & \zeta & \delta & \epsilon & \\ & & & & \ddots & \ddots & \ddots \\ & & & & & \zeta & \delta \end{pmatrix}, \quad Q = \begin{pmatrix} \delta & & & & & & \\ \zeta & \delta & \epsilon & & & & \\ & & \delta & & & & \\ & & \zeta & \delta & \epsilon & & \\ & & & & \delta & & \\ & & & & \ddots & \ddots & \ddots \\ & & & & & & \delta \end{pmatrix},$$

ce sont des matrices de taille $n \times n$. La matrice du problème est une matrice $mn \times mn$ donnée par

$$A = \begin{pmatrix} L & P & & & & & \\ P^T & M & Q & & & & \\ & Q^T & L & P & & & \\ & & P^T & M & Q & & \\ & & & Q^T & L & P & \\ & & & & \ddots & \ddots & \ddots \\ & & & & & Q & L \end{pmatrix}.$$

C'est une matrice de type TBT. En effet, si on pose

$$Z^{(g)} = \begin{pmatrix} 0_{2n\times 2n} & & & \\ I_{2n} & \ddots & & \\ & \ddots & \ddots & \\ & & R & 0_{2n\times 2n} \end{pmatrix},$$

avec $R = (I_n \, 0_{2n\times 2n})$, et

$$Z^{(d)} = I_n \otimes Z_n^2$$

alors on peut vérifier que

$$\text{rang}(A - Z^{(g)} A Z^{(g)T}) \leq 4n$$

et

$$\text{rang}(A - Z^{(d)} A Z^{(d)T}) \leq 4m.$$

Ce cas n'est pas traité par les algorithmes présentés dans ce travail, mais il a une structure très intéressante.

Exemple 3

Discrétisation $\mathbb{P}_2$ avec maillage triangle rectangle uniforme. Posons :

$$\lambda_1 = \mathcal{A}(\varphi_1, \varphi_1),\ \lambda_2 = \mathcal{A}(\varphi_2, \varphi_2),\ \lambda_3 = \mathcal{A}(\varphi_1, \varphi_2),$$
$$\mu_1 = \mathcal{A}(\varphi_1, \varphi_{n+1}),\ \mu_2 = \mathcal{A}(\varphi_2, \varphi_{n+2}),\ \mu_3 = \mathcal{A}(\varphi_1, \varphi_{n+2}),\ \mu_4 = \mathcal{A}(\varphi_3, \varphi_{n+2}),$$
$$\mu_5 = \mathcal{A}(\varphi_2, \varphi_{n+3}),\ \mu_6 = \mathcal{A}(\varphi_2, \varphi_{n+4})$$
$$\rho_1 = \mathcal{A}(\varphi_{n+2}, \varphi_{3n+3}),\ \rho_2 = \mathcal{A}(\varphi_{n+2}, \varphi_{3n+2}),\ \rho_3 = \mathcal{A}(\varphi_{n+2}, \varphi_{3n+4}),$$
$$\pi_1 = \mathcal{A}(\varphi_{n+1}, \varphi_{n+1}),\ \pi_2 = \mathcal{A}(\varphi_{n+2}, \varphi_{n+2}),\ \pi_3 = \mathcal{A}(\varphi_{n+2}, \varphi_{n+4}),\ \pi_4 = \mathcal{A}(\varphi_{n+1}, \varphi_{n+2}),$$
$$\chi_1 = \mathcal{A}(\varphi_{n+1}, \varphi_{2n+1}),\ \chi_2 = \mathcal{A}(\varphi_{n+1}, \varphi_{2n+2}),\ \chi_3 = \mathcal{A}(\varphi_{n+3}, \varphi_{2n+2}),\ \chi_4 = \mathcal{A}(\varphi_{n+2}, \varphi_{2n+4})$$

et

$$L = \begin{pmatrix} \lambda_1 & \lambda_3 & & & & \\ \lambda_3 & \lambda_2 & \lambda_3 & & & \\ & \lambda_3 & \lambda_1 & \lambda_3 & & \\ & & \lambda_3 & \lambda_2 & \lambda_3 & \\ & & & \ddots & \ddots & \ddots \\ & & & & \lambda_3 & \lambda_1 \end{pmatrix}, \quad M = \begin{pmatrix} \mu_1 & \mu_3 & & & & & \\ & \mu_2 & \mu_5 & \mu_6 & & & \\ & \mu_4 & \mu_1 & \mu_3 & & & \\ & & & \mu_2 & \mu_5 & \mu_6 & \\ & & & \ddots & \ddots & \ddots & \\ & & & & & \mu_4 & \mu_1 \end{pmatrix},$$

$$P = \begin{pmatrix} \pi_1 & \pi_4 & & & & & \\ \pi_4 & \pi_2 & \pi_4 & \pi_3 & & & \\ & \pi_4 & \pi_1 & \pi_4 & & & \\ & \pi_3 & \pi_4 & \pi_1 & \pi_4 & \pi_3 & \\ & & & \ddots & \ddots & \ddots & \\ & & & & & \pi_4 & \pi_1 \end{pmatrix}, \quad Q = \begin{pmatrix} \chi_1 & \chi_2 & & & & & \\ \mu_4 & \mu_2 & \mu_3 & \chi_4 & & & \\ & \chi_3 & \chi_1 & \chi_2 & & & \\ & & \mu_4 & \mu_2 & \mu_3 & \chi_4 & \\ & & & \ddots & \ddots & \ddots & \\ & & & & & \chi_3 & \chi_1 \end{pmatrix},$$

$$R = \begin{pmatrix} \rho_1 & & & & & \\ \rho_2 & \rho_1 & \rho_3 & & & \\ & & \rho_1 & & & \\ & & \rho_2 & \rho_1 & \rho_3 & \\ & & & \ddots & \ddots & \ddots \\ & & & & & \rho_1 \end{pmatrix}, \quad A = \begin{pmatrix} L & M & & & & \\ M^T & P & Q & R & & \\ & Q^T & L & M & & \\ & R^T & M^T & P & Q & R \\ & & & \ddots & \ddots & \ddots \\ & & & & Q^T & L \end{pmatrix}.$$

Dans ce cas, les opérateurs de déplacement définis moyen de

$$Z^{(g)} = Z_m^2 \otimes I_n, \quad Z^{(d)} = I_m \otimes Z_n^2$$

vont conduire aux observations suivantes :

$$\text{rang}(A - Z^{(g)} A Z^{(g)T}) \leq 4n,$$

et

$$\text{rang}(A - Z^{(d)} A Z^{(d)T}) \leq 4m.$$

3.3 Résolution de systèmes d'équations polynomiales

Les matrices structurées en multiniveaux sont naturellement associées aux systèmes d'équations polynomiales multivariées. En effet, les matrices de résultant, dont leur déterminant est le résultant, représente la méthode la plus puissante pour calculer le résultant et pour résoudre les systèmes d'équations polynomiales. Parmi ces matrices, on spécifie la matrices de Sylvester, qui est, par construction, une matrice de Toeplitz multiniveau, et la matrice de Bézout qui est l'inverse d'une matrice de Hankel, voir [88]. Les références [87], [13], [19] décrivent comment utiliser les matrices de résultant pour transformer le problème de résolutions de systèmes polynomiaux à un problème linéaire.

Les systèmes polynomiaux, et par suite les matrices de Toeplitz et de Hankel multiniveaux, parraissent dans plusieurs applications de robotiques [20], [109], dans la structure

géométrique des molécules [38], dans la résolution des systèmes d'inégalités [55] et plusieurs d'autres applications.

Prenons, par exemple, le problème de [13] : Pour un système d'équations polynomiales, on cherche, parmi les racines de ce système, la racine qui maximise (resp. minimise) la valeur absolue d'un polynôme fixé. On peut réduire ce problème à un problème de calcul de la plus grande (resp. la plus petite) valeur propre d'une matrice A qui est une matrice de Toeplitz multiniveau voir [13]. Pour calculer la plus grande valeur de A on utilise la méthode de la puissance. Cette méthode ne demande que la multiplication de A par un vecteur, donc chaque itération demande $\mathcal{O}(N \log^2 N)$ à la place de $\mathcal{O}(N^2)$ flops, avec $N \times N$ est la taille de la matrice A. Le calcul de la plus petite valeur propre se fait à l'aide de la méthode de puissance inverse. A chaque itération de cet algorithme on doit résoudre un système linéaire correspond à la matrice A. D'ici, vient l'importance de trouver une méthode de résolution rapide.

3.4 Traitement d'images numériques et du signal

On désigne par traitement d'images l'ensemble des techniques permettant de modifier une image numérique dans le but de l'améliorer ou d'en extraire des informations.

Les matrices de Toeplitz par blocs de Toeplitz paraissent dans la plupart de ces techniques, on va choisir quelques uns.

Dans le traitement numérique d'images, il est souvent nécessaire de procéder à des transformations géométriques non linéaires des positions des pixels de la grille de l'image. Les transformées des positions des pixels ne coïncident alors généralement pas avec la grille de l'image d'entrée. Un algorithme de ré-échantillonnage permet d'obtenir la valeur de ces nouveaux pixels. Un algorithme de ré-échantillonnage par la B-spline cubique uniforme est donné dans [80]. L'écriture matricielle de cet algorithme est

$$G' = BCB^T$$

avec G' la matrice de l'image ré-échantionnée, B une matrice TBT et C une matrice correspond à la l'image d'entrée.

Le problème de restauration d'une image peut se transformer en un problème de résolution d'un système linéaire $Tx = b$ avec T est une matrice TBT de taille $n^2 \times n^2$ pour une image formée de $n \times n$ pixels, voir [76]. Vu la grande taille de ces matrices, un algorithme de résolution rapide pour un système TBT est crucial. Prenons l'exemple donné dans cet article. On considère le problème d'astronomie suivant : on dispose d'une image observée par un satellite et on veut reconstruire cette image. Ce problème se réduit en un système linéaire $Hx = b$ où H est la fonction de flou qui peu être estimer et b est un vecteur donné par l'observation du satellite. Cependant H est une matrice rectangulaire et circulante par blocs circulants. Donc, on doit résoudre, à la place de l'équation initiale, l'équation normale $H^*Hx = H^*b$. Comme la matrice H^*H est malle conditionnée, on utilise la régularisation suivante :

$$(\lambda I + H^*H)x = H^*b$$

où λ est le paramètre de régularisation. La matrice $\lambda I + H^*H$ est une matrice TBT de taille $n^2 \times n^2$. Voir [76].

Chapitre 4

Peut-on trouver des méthodes de résolution rapide pour Toeplitz par blocs de Toeplitz ?

4.1 Introduction

Malgré un progrès remarquable en étude des algorithmes de résolution rapides pour les matrices structurées scalaires ces dernières décennies, les matrices structurées multiniveaux, en particulier les matrices de Toeplitz par blocs de Toeplitz (TBT), restent un défi. On a donné dans le premier chapitre beaucoup de références qui traitent du cas scalaire, alors que le cas par blocs est très peu étudié. La difficulté de ce problème explique le faible nombre d'études qui portent sur la résolution des systèmes TBT. L'étude des matrices TBT a commencé en même temps que l'étude des matrices de Toeplitz scalaires au début des années 80. Comme les applications conduisent à des matrices de grande taille, un algorithme de résolution rapide pour de telles matrices est crucial. La seule publication qui essaie de résoudre ce problème est [96]. Elle propose d'approcher l'inverse d'une matrice TBT par une somme de produits de Kronecker de matrices de type Toeplitz. Cet angle d'attaque est purement expérimental et, à ce jour, il n'existe pas de résultats théoriques décrivant un algorithme rapide de résolution pour des systèmes TBT généraux.

Les algorithmes utilisés pour le cas scalaire ne s'adaptent pas facilement au cas d'une matrice TBT. On peut prendre comme exemple la fameuse formule de Gohberg-Semencul, qui n'existe pas pour les matrices TBT.

Notons aussi que les méthodes itératives qui utilisent des préconditionneurs superlinéaire rencontrent plusieurs problèmes : premièrement en [116], les auteurs montrent que les préconditionneurs de type circulant ne sont pas superlinéaires, puis que de tels préconditionneurs ne peuvent pas appartenir à l'algèbredes matrices de Toeplitz symétriques par blocs de Toeplitz symétriques ni à l'algèbre des matrices circulantes par blocs circulants, [93]. Un résultat plus négatif est donné en [117]. Ces résultats indiquent que la construction de préconditionneurs efficaces n'est pas possible dans une classe encore plus vaste d'algèbres de matrices.

On remarquera, dans le chapitre 5, un comportement bizarre des matrices Toeplitz

bande par blocs Toeplitz bande, qu'on peut résumer ainsi : si on prend des matrices (pseudo-)aléatoires dans cette classe, une statistique expérimentale montre qu'elles sont mal conditionnées, et donc le système correspondant difficile à résoudre.

De là on pose la question : peut-on résoudre rapidement, en $\mathcal{O}(N \log^{\omega} N)$, un système à matrice de Toeplitz par blocs de Toeplitz, et en exploitant les deux structures en même temps ? Ici $N \times N$ est la taille de la matrice et $\omega \in \mathbb{R}$.

Avant de répondre à cette question, on étudiera le cas d'une matrice de Toeplitz par blocs dans la section suivante. Dans la troisième section on va essayer d'appliquer les techniques de la deuxième section aux matrices de Toeplitz par blocs de Toeplitz, et on étudiera les propriétés particulières aux matrices de Toeplitz par blocs de Toeplitz ainsi que leurs déplacements. Dans la dernière section, on étudiera quelques cas particuliers et on donnera quelques idées de résolutions.

Tout d'abord, donnons quelques définitions et rappelons brièvement la multiplication de Kronecker et quelques uns de ses propriétés.

Définition 4.1.1. *On notera $P_{m,n}$, ou tout simplement P s'il y a pas de confusion, la matrice de permutation suivante, de taille $mn \times mn$:*

$$P = \begin{pmatrix} E_1 \\ \vdots \\ E_m \end{pmatrix} \quad \text{avec } E_k = \begin{pmatrix} e_k^T \\ e_{k+m}^T \\ \vdots \\ e_{k+(n-1)m}^T \end{pmatrix}. \tag{4.1}$$

Définition 4.1.2. *Soient A et B deux matrices de terme général a_{ij} et b_{ij}, de taille $m \times n$ et $p \times q$ respectivement. Le produit de Kronecker, $A \otimes B$, de A par B est la matrice, de taille $mp \times nq$, suivante :*

$$A \otimes B = \begin{pmatrix} a_{1,1}B & \dots & a_{1,m}B \\ \vdots & \ddots & \vdots \\ a_{n,1}B & \dots & a_{n,m}B \end{pmatrix}$$

Proposition 4.1.3. *Soient A, B, C et D quatre matrices. On a les propriétés suivantes :*

- *Pour tout α dans $\mathbb{K}$,*

$$A \otimes (\alpha B) = \alpha(A \otimes B) = (\alpha A) \otimes B.$$

- *Le produit de Kronecker est distributif par rapport à l'addition :*

$$(A + B) \otimes C = (A \otimes C) + (B \otimes C),$$
$$A \otimes (B + C) = (A \otimes B) + (A \otimes C).$$

- *Le produit de Kronecker est associatif :*

$$(A \otimes B) \otimes C = A \otimes (B \otimes C).$$

- *La transposition est donnée par la formule ci-dessous. Remarquons la conservation de l'ordre :*

$$(A \otimes B)^T = A^T \otimes B^T.$$

– *Multiplication, quand les dimensions sont compatibles :*

$$(A \otimes B)(C \otimes D) = (AC \otimes BD).$$

– *Quand A et B sont carrées et inversibles, alors :*

$$(A \otimes B)^{-1} = (A^{-1} \otimes B^{-1}).$$

– *Spectre : Soit $A \in \mathbb{K}^{n\times n}$ et $B \in \mathbb{K}^{m\times m}$. Si $\lambda_1, \ldots, \lambda_n$ et $\mu_1, \ldots, \mu_m$ sont les valeurs propres de A et B respectivement, alors $\{\lambda_i\mu_j,\ i = 1, \ldots, n$ et $j = 1, \ldots, m\}$ sont les valeurs propres de $A \otimes B$ (en comptant la multiplicité). En particulier :*

$$\begin{aligned} &\det(A \otimes B) = \det(A)^m.\det(B)^n, \\ &\mathit{trace}(A \otimes B) = \mathit{trace}(A).\mathit{trace}(B), \\ &\mathit{rang}(A \otimes B) = \mathit{rang}(A)\mathit{rang}(B). \end{aligned}$$

Démonstration. Voir [54] pour plus d'information sur le produit de Kronecker de deux matrices. □

Proposition 4.1.4. *Soient $A \in \mathbb{K}^{n\times n}$, $B \in \mathbb{K}^{m\times m}$. La multiplication de $M = A \otimes B$ par un vecteur v de longueur mn coûte $\mathcal{O}(mC_A + nC_B)$ flops, avec, pour une matrice D, C_D est le coût de la multiplication de D par un vecteur.*

Démonstration. On décompose v en blocs vectoriels de taille m, on l'écrit donc

$$v = \begin{pmatrix} v_1 \\ \vdots \\ v_n \end{pmatrix}.$$

On a donc

$$Mv = (A \otimes I_m) \begin{pmatrix} w_1 \\ \vdots \\ w_n \end{pmatrix},$$

avec $w = (w_i := Bv_i)_i$. Or $P(A \otimes I_m)P^T = I_m \otimes A$, où P est la matrice définie en (4.1). Donc, $(A\otimes I_m)w = P^T(I_m\otimes A)Pw = \begin{pmatrix} Aw'_1 \\ \vdots \\ Aw'_m \end{pmatrix}$, avec w' le vecteur Pw décomposé en blocs vectoriels de taille n. Ce qui donne à la fin n multiplications par B et m multiplications par A. □

4.2 Matrices de Toeplitz par blocs

Soit T une matrice de Toeplitz par blocs formée de m blocs de taille $n \times n$ chacun. Donc, T est de la forme suivante :

$$T = \begin{pmatrix} T_0 & T_{-1} & \ldots & T_{-m+1} \\ T_1 & T_0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & T_{-1} \\ T_{-m+1} & \ldots & T_1 & T_0 \end{pmatrix}, \tag{4.2}$$

où chaque T_i, $i = -m+1, \ldots, m-1$, est une matrice de taille $n \times n$.

Une définition du déplacement par blocs est possible, en prenant les opérateurs de déplacement, $D_{Z_\varphi \otimes I_n, Z_\psi \otimes I_n}$ et $\Delta_{Z_\varphi \otimes I_n, Z_\psi^T \otimes I_n}$. Donc, une généralisation directe des algorithmes de résolution, rapide et ultra-rapide, des systèmes de Toeplitz est possible. Dans [137] les auteurs font une généralisation de l'algorithme de Levinson et de l'algorithme de Schur, puis ils généralisent un algorithme ultra-rapide. Les algorithmes rapides donnés dans cet article demandent $\mathcal{O}(n^3m^2)$ flops, et l'algorithme ultra-rapide demande $\mathcal{O}(n^3m + n^2m\log^2 m)$ flops. Reprenons, par exemple, l'algorithme de Levinson, pour voir comment faire la généralisation et pour remarquer la simplicité de cette généralisation. On reprend l'algorithme de Levinson décrit dans le premier chapitre avec toutes les notations définies pour cet algorithme :

Ici les t_i, les y_i, les z_i et les s_i sont des blocs de taille $n \times n$, les u_i et les b_i sont des vecteurs de longueur n. Le même calcul que dans le cas scalaire nous donne :

$$U_{k+1} = \begin{pmatrix} U_k - u_{k+1}\hat{Y}_k \\ u_{k+1} \end{pmatrix}, \text{ avec } \begin{array}{l} u_{k+1} = s_k^{-1}(-\hat{W}_k^T U_k + b_{k+1}), \\ U_1 = u_1 = t_0^{-1} b_1 \end{array}. \tag{4.3}$$

$$Y_{k+1}^T = \begin{pmatrix} Y_k^T - y_{k+1}\hat{Z}_k^T \\ y_{k+1} \end{pmatrix}, \text{ avec } \begin{array}{l} y_{k+1} = s_k^{-1}(-Y_k^T \hat{V}_k + t_{-k-1}), \\ Y_1 = y_1 = t_0^{-1} t_{-1} \end{array}. \tag{4.4}$$

$$Z_{k+1} = \begin{pmatrix} Z_k - z_{k+1}\hat{Y}_k \\ z_{k+1} \end{pmatrix}, \text{ avec } z_{k+1} = s_k^{-1}(-\hat{W}_k^T Z_k + t_{k+1}), \text{ et } Z_1 = t_0 t_1. \tag{4.5}$$

Et

$$s_{k+1} = s_k(I_n + z_{k+1}^T y_{k+1}).$$

Nombre d'opérations :

$3n^3 + n^2$ flops pour calculer s_k.

$n^3 + 2n^2 + n$ flops pour calculer u_k, et $kn^2 + kn$ flops pour calculer ν_k.

$kn^3 + 2n^3 + n^2$ flops pour calculer y_k (resp. z_k), et $kn^3 + kn^2$ flops pour calculer μ_k (resp. β_k).

Ceci donne un coût total en $\mathcal{O}(n^3m^2)$ flops. On remarque que si la matrice T est TBT alors le coût de calcul des y_k (resp z_k) sera $kn^2log^2n + 2n^3 + n^2$ flops si on utilise un algorithme de multiplication rapide.

De fait, les algorithmes classiquespour les matices de type Toeplitz donnent au moins aussi bien. En effet, la matrice $T' = PTP^T$ est une matrice par blocs de Toeplitz, elle est formée de n blocs de Toeplitz de taille $m \times m$ chacun. On peut remarquer facilement que la matrice T' est une matrice de type Toeplitz scalaire de rang de déplacement au plus $r = 2n$. En utilisant les algorithmes associés aux matrices de type Toeplitz, on obtient des algorithmes de résolution rapide qui demandent $\mathcal{O}(r(nm)^2) = \mathcal{O}(n^3m^2)$ flops et on obtient des algorithmes de résolution ultra-rapide qui demandent $\mathcal{O}(r^2(mn)\log^2 mn) = \mathcal{O}(n^3m\log^2 nm)$ flops.

Récemment, un nouvel algorithme, donné dans [15], résout un système de type Toeplitz de rang de déplacement r en $\mathcal{O}(r^{\omega-1}n\log^2 n)$, avec $\mathcal{O}(n^\omega)$ l'ordre de complexité de la multiplication de deux matrices de taille $n \times n$. La meilleure estimation actuelle d'ω

est 2.38. En appliquant cet algorithme à T' on obtient un algorithme de résolution en $\mathcal{O}(n^\omega m \log^2 mn)$.

Dans le cas des matrices de Toeplitz par blocs, on a su profiter de la structure par blocs pour donner des algorithmes de résolution rapide. En passant aux matrices de Toeplitz par blocs de Toeplitz, comment profite-on, de plus, de la structure des blocs ? Est-ce possible ?

4.3 Matrices de Toeplitz par blocs de Toeplitz

Dans la suite, sauf si elle est définie autrement, T désigne une matrice de Toeplitz par blocs de Toeplitz formée de m blocs de taille $n \times n$ chacun. T est de la forme (4.2) et de plus chaque bloc T_i, pour $i = -m+1, \ldots, m-1$, est une matrice de Toeplitz scalaire.

On commence par donner trois propriétés simples et importantes d'une matrice de Toeplitz par blocs de Toeplitz.

Une matrice de Toeplitz scalaire est une matrice symétrique par rapport à l'antidiagonale. Ceci se traduit par :

$$JT^TJ = T.$$

En effet, JT est une matrice de Hankel, donc symétrique. Ainsi $JT = T^TJ$ et parsuite $JT^TJ = T$. Une matrice de Toeplitz par blocs est donc symétrique, par blocs, par rapport à l'antidiagonale par blocs. Mais pour les matrices de Toeplitz par blocs de Toeplitz, on peut faire mieux :

Proposition 4.3.1. *Soit T une matrice de Toeplitz par blocs de Toeplitz. T est symétrique par rapport à l'antidiagonale.*

Démonstration. Soit J la matrice qui ne contient que des 1 sur l'antidiagonale de taille $mn \times mn$, et soit J_m et J_n la même matrice de taille $m \times m$ et $n \times n$ respectivement. On remarque que $J = J_m \otimes J_n$, donc

$$\begin{aligned} JTJ &= \begin{pmatrix} J_nT_0J_n & \cdots & J_nT_{n-1}J_n \\ \vdots & \ddots & \vdots \\ J_nT_{-m+1}J_n & \cdots & J_nT_0J_n \end{pmatrix} \\ &= \begin{pmatrix} T_0^T & \cdots & T_{m-1}^T \\ \vdots & \ddots & \vdots \\ T_{-m+1}^T & \cdots & T_0^T \end{pmatrix} = T^T \end{aligned}$$

□

Remarque 4.3.2. *L'inverse de T est aussi symétrique par rapport à l'antidiagonale.*

La deuxième propriété de la matrice de Toeplitz par blocs de Toeplitz T est la possibilité de la transformer, par la permutation P, définie en (4.1), en une autre matrice TBT formée de n blocs de taille $m \times m$ chacun. Cette propriété sera très utile si $m \ll n$. En effet, on peut résoudre un système de Toeplitz par blocs, et par suite de Toeplitz par blocs de Toeplitz, en $\mathcal{O}(n^3 m \log^2(nm))$, donc ces algorithmes seront très pratiques si $n \ll m$.

Inversement si $m \ll n$, ils seront très peu pratiques dans le cas d'une matrice de Toeplitz par blocs, mais ils resteront très pratiques pour le cas d'une matrice TBT, parce qu'on peut la transformer en une autre matrice de Toeplitz par blocs où on échange les rôles de m et de n.

Proposition 4.3.3. *La matrice $T' = PTP^T$ est une matrice TBT formée de n blocs $m \times m$ chacun, et elle est de la forme suivante :*

$$T' = \begin{pmatrix} T'_0 & \dots & T'_{-n+1} \\ \vdots & \ddots & \vdots \\ T'_{n-1} & \dots & T'_0 \end{pmatrix}$$

avec

$$T'_j = \begin{pmatrix} t_{j,0} & \dots & t_{j,-m+1} \\ \vdots & \ddots & \vdots \\ t_{j,m-1} & \dots & t_{j,0} \end{pmatrix},$$

pour $-n+1 \leq j \leq n-1$.

Démonstration. Si on indexe une matrice par blocs M, formée de m blocs et les blocs sont de taille $n \times n$, de la manière suivante :

$$M = \left(M_{(i_1,i_2),(j_1,j_2)}\right)_{\substack{0 \leq i_1, j_1 \leq m-1 \\ 0 \leq i_2, j_2 \leq n-1}}, \tag{4.6}$$

(i_1, j_1) donnant les positions des blocs, (i_2, j_2) donnant les positions dans les blocs, alors, comme T est une matrice de Toeplitz par blocs de Toeplitz, on peut l'écrire

$$T = (t_{(i_1-j_1,i_2-j_2)})_{\substack{0 \leq i_1, j_1 \leq m-1. \\ 0 \leq i_2, j_2 \leq n-1}}$$

On remarque facilement que la matrice PTP^T a $t_{(i_2-j_2,i_1-j_1)}$ comme terme général. □

La troisième propriété importante d'une matrice de Toeplitz par blocs de Toeplitz est la multiplication rapide par un vecteur :

Proposition 4.3.4. *Soient T une matrice de Toeplitz par blocs de Toeplitz de taille $mn \times mn$ et v un vecteur de longueur mn. On peut calculer Tv en utilisant $\mathcal{O}(mn \log mn)$ flops.*

Démonstration. Voir le corollaire (4.4.4). □

4.3.1 Le déplacement en deux dimensions

Dans la suite de ce chapitre, pour $k \in \mathbb{N}$, Z_k désigne la matrice Z de shift vers le bas, donnée par la définition (2.1.3) de taille $k \times k$ avec $\varphi = 0$. Pour d'autres $\varphi \in \mathbb{K}$, on pourrait faire presque le même calcul.

La matrice T est de Toeplitz par blocs de Toeplitz ; en particulier c'est une matrice par blocs de Toeplitz, donc, comme on l'a vu précédemment, c'est une matrice de type Toeplitz scalaire, de rang de déplacement $2n$. Ce déplacement scalaire n'exploite que la structure en blocs. Pour faire mieux, on passe au déplacement par blocs :

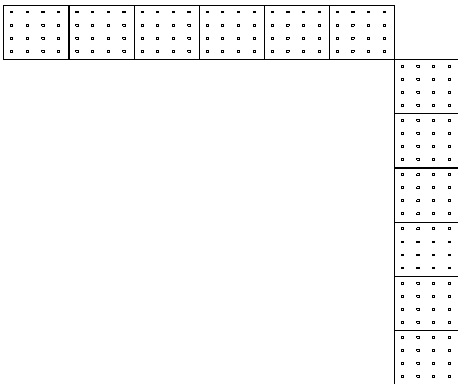


FIG. 4.1 – La structure de $D_1(T)$ pour $m = 7$ et $n = 4$. On remarque que les blocs de cette matrice sont tous nuls sauf dans la première ligne et dernière colonne. Donc $\text{rang}(D_1(T)) = 2n$.

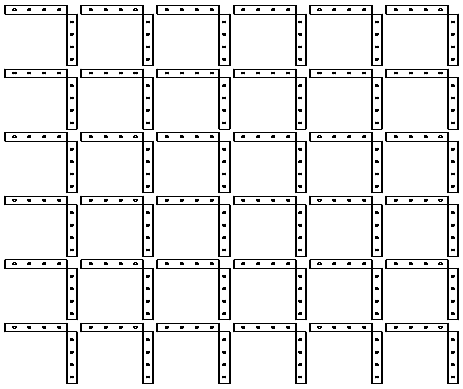


FIG. 4.2 – La structure de $D_2(T)$ pour $m = 6$ et $n = 5$. On remarque que cette matrice contient seulement m lignes et m colonnes non nuls. Donc $\text{rang}(D_2(T)) = 2m$.

Définition 4.3.5. *On note $Z^{(g)}$ et $Z^{(d)}$ les deux matrices, de taille $mn \times mn$, données par :*

$$Z^{(g)} = Z_m \otimes I_n, \tag{4.7}$$
$$Z^{(d)} = I_m \otimes Z_n. \tag{4.8}$$

On définit les deux opérateurs de déplacement suivants :

Définition 4.3.6. *Soit A une matrice par blocs de taille $mn \times mn$. On définit D_1 et D_2 de la manière suivante :*

$$D_1(A) = D_{Z^{(g)},Z^{(g)}}(A) = Z^{(g)}A - AZ^{(g)} \tag{4.9}$$
$$D_2(A) = D_{Z^{(d)},Z^{(d)}}(A) = Z^{(d)}A - AZ^{(d)} \tag{4.10}$$

Les opérateurs D_1 et D_2 correspondent respectivement au déplacement par blocs et au déplacement à l'intérieur des blocs. On remarque que $\text{rang}(D_1(T)) = 2n$ et $\text{rang}(D_2(T)) = 2m$, on se reportera aux figures. (4.1) et (4.2), Le calcul avec ces opérateurs se fait de la même manière qu'avec l'opérateur scalaire $D_{Z,Z}$: D_1 et D_2 ont les mêmes propriétés que D, en particuliers $\text{rang}(D_1(T^{-1})) = 2n$ et $\text{rang}(D_2(T^{-1})) = 2m$, et des propriétés analogues ont lieu pour un complément de Schur dans T. On peut donc généraliser les algorithmes associés aux matrices de Toeplitz scalaires, en utilisant la déplacement $D_1(T)$ ou $D_2(T)$ chacun à part, c'est-à-dire en exploitant une seule structure de T seulement.

La question évidente est donc : est ce qu'en prenant le double déplacement $D = D_1 \circ D_2$, qui est égal à $D_2 \circ D_1$, obtient-on un bon déplacement de T qui exploite les deux structures et qui satisfait les trois conditions d'un opérateur de déplacement ? La réponse est, malheureusement, non ! Et de plus cet opérateur complique beaucoup le calcul sans donner des avantages par rapport à D_1 ou D_2.

Proposition 4.3.7. *Le D-rang de déplacement de T est au plus* $2\min(m, n)$.

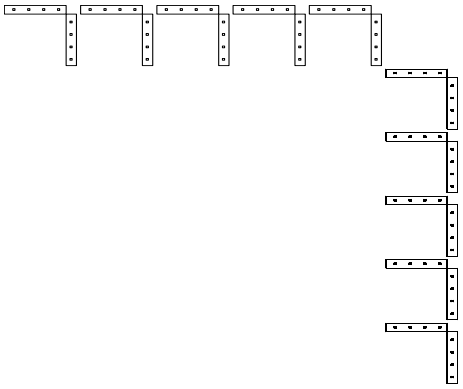


FIG. 4.3 – La structure de $D(T)$ pour $m = 6$ et $n = 5$. $D(T)$ est très creuse mais elle est de rang $2\min(n, m)$.

Démonstration. Tous les blocs de la matrice par blocs $D(T)$ sont nuls sauf ceux de la première ligne et de la dernière colonne. Donc $\text{rang}(D(T)) \leq 2n$. Si $n \geq m$, alors $PD(T)P^T$ est de rang au plus $2n$ pour la même raison, parsuite $\text{rang}(D(T)) \leq 2m$. Voir la figure (4.3), □

On remarque tout d'abord que $D(T)$ (resp. $D_1(T)$ et $D_2(T)$) contient toutes les informations de T.

Dans le cas générique, si T ne contient pas de zéros et s'il n'y a pas de coefficient répété dans sa première ligne et dans sa première colonne, le D-rang de déplacement de T est $2\min(m, n)$. Donc, on ne gagne rien du point de vue du rang en utilisant D et non pas D_1 si $m \leq n$ ou D_2 dans le cas contraire.

Certes $D(T)$ est beaucoup plus creuse que $D_1(T)$ et $D_2(T)$, et on peut donner des générateurs de $D(T)$ dont les blocs, de taille $n \times n$, sont de rang 2. On remarque de plus que les valeurs singulières de $D(T)$ décroissent plus vite que les valeurs singulières de $D_1(T)$ ou $D_2(T)$. Voir la figue (4.4). Est ce que cette propriété est une avantage pour le calcul ? On prétend que non, parce que T^{-1} (resp. un complément de Schur de T) n'a pas une telle propriété. Parce que $D(T^{-1})$ n'est pas creuse et ces blocs ne sont pas structurées, et de plus le D-rang de déplacement de T^{-1} n'égale pas, en général, le D-rang de déplacement de T, mais plutôt son double, comme le montre la proposition (4.3.8). On remarque aussi que les valeurs singulières de $D(T^{-1})$ décroissent moins vite que les valeurs singulières de $D_1(T^{-1})$ et de $D_2(T^{-1})$, voir figure (4.5), qui met en évidence un effet de rang.

Proposition 4.3.8. *Si T est inversible, alors rang$(D(T^{-1}))$ est au plus $4min(m, n)$.*

Démonstration. Supposons que $m \leq n$. On a $D_2(T) = Z^{(d)}T - TZ^{(d)}$; en multipliant cette équation par T^{-1} gauche et à droite, il vient $D_2(T^{-1}) = -T^{-1}D_2(T)T^{-1}$. Alors $\text{rang}(D_2(T^{-1}) = \text{rang}(D_2(T)) = 2m$. Comme $D(T^{-1}) = D_1(D_2(T^{-1})) = Z^{(d)}D_2(T^{-1}) - D_2(T^{-1})Z^{(d)}$ alors $\text{rang}(D(T))$ est au plus $2m + 2m = 4m$. □

Remarque 4.3.9. *1. Des exemples simples prouvent qu'en général, le D-rang de déplacement est effectivement $4min(m, n)$.*

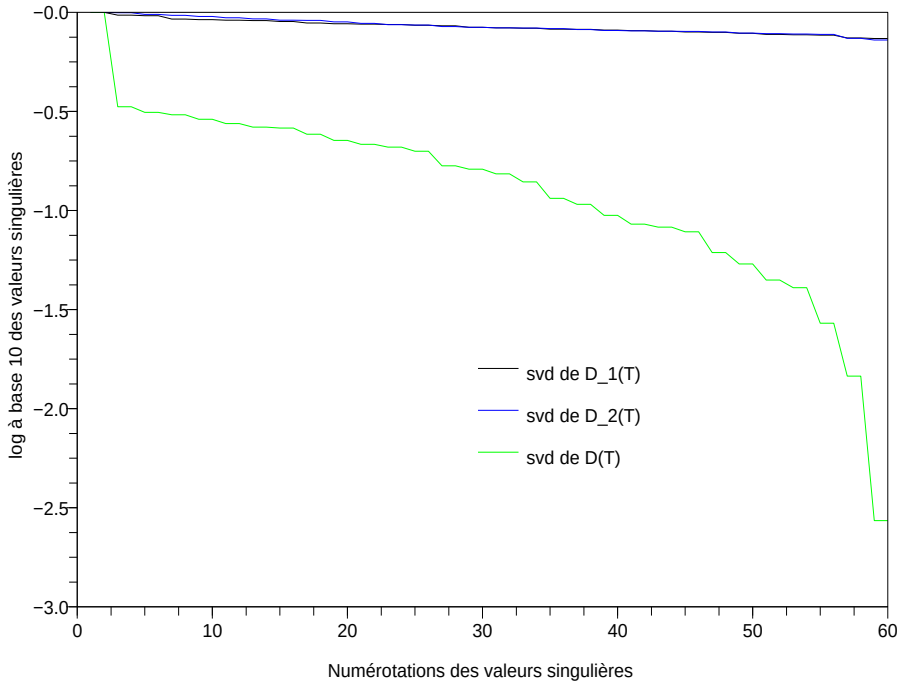


FIG. 4.4 – Les valeurs singulières de $D_1(T)$, $D_2(T)$ et $D(T)$. La matrice T est TBT, elle est choisie au hasard avec $m = n = 30$.

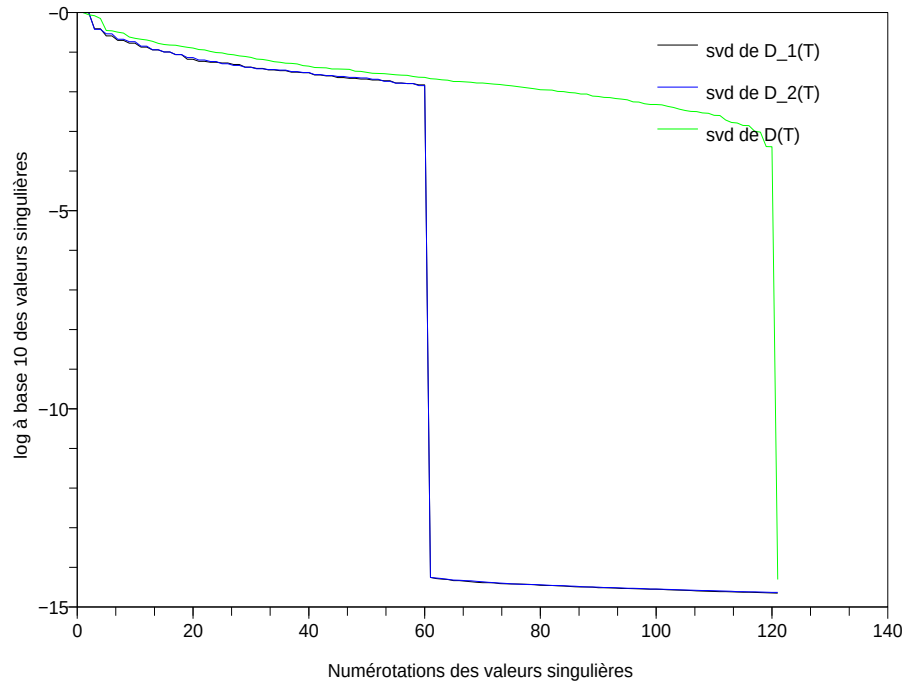


FIG. 4.5 – Les valeurs singulières de $D_1(T^{-1})$, $D_2(T^{-1})$ et $D(T^{-1})$. La matrice T est TBT, elle est choisie au hasard avec $m = n = 30$.

2. *Une conséquence directe de la proposition précédente et de la première remarque est que le D-rang de déplacement d'un complément de Schur dans une matrice TBT est en général, plus grand que le D-rang de déplacement de T.*

De plus, la reconstruction d'une matrice à partir de son déplacement par D n'est pas une chose évidente et simple et elle demande beaucoup plus d'opérations que le cas scalaire.

Donc en se basant sur les trois critères, énoncés page 21, qu'un opérateur de déplacement doit vérifier, D ne fournit pas un bon déplacement pour une matrice de type Toeplitz à deux niveaux.

En suivant [114], on peut essayer de définir des matrices de type TBT par les conditions $\text{rang}(D_1(T)) \leq r_1 n$ et $\text{rang}(D_2(T)) \leq r_2 m$, avec $r_1 \ll n$ et $r_2 \ll m$. Les deux questions qui se posent ici sont donc :

1. Est ce qu'il y a des opérateurs de déplacement qui exploitent la structure d'une matrice de type Toeplitz à deux niveaux ?
2. Est ce que dans le cas de structures à deux niveaux, il faudrait changer les critères de qualification d'un opérateur de déplacement, en envisageant éventuellement d'autres propriétés que "petit rang" ?

Dans le cas scalaire, on a profité de la structure de déplacement des matrices structurées pour développer des techniques de calcul. Ces techniques ont mené à des algorithmes de résolution rapide pour des systèmes structurés. Dans le cas de déplacement à deux niveaux, on ne sait pas répondre aux deux questions posés ci-dessus. Comme on sait pas répondre à la première question, on ne peut pas profiter de ces techniques en deux dimensions. Et sans réponseà la deuxième question, on ne voit pas comment généraliser les techniques du cas scalaire au cas à deux niveaux.

Pour les matrices de Toeplitz, on a pu donner des algorithmes de résolution rapide et ultra-rapide, même avant de définir la structure de déplacement. Peut-on faire de même dans le cas d'une matrice de Toeplitz par blocs de Toeplitz, c'est-à-dire d'essayer d'utiliser la structure de Toeplitz de la matrice et non pas sa structure de déplacement ? La réponse est probablement non, parce que les blocs de T^{-1} perdent la structure, son rang de déplacement par $D_{Z,Z}$ est maximal. Cependant, T^{-1} est de type Teoplitz en blocs, c'est-à-dire $D_{Z^{(d)},Z^{(d)}}(T^{-1})$ est de rang $2m$.

4.4 Cas particuliers et quelques idées de résolution

Il y a des cas particuliers de matrice TBT où on sait résoudre rapidement. Les matrices circulantes par blocs circulants, circulantes par blocs de Toeplitz ou Toeplitz triangulaire par blocs de Toeplitz triangulaires possèdent, comme on va voir, des algorithmes de résolution rapide. On va étudier le cas d'une matrice de Toeplitz bande par blocs de Toeplitz bande dans le chapitre (5).

4.4.1 Matrice circulante par blocs circulants

Définition 4.4.1. *Soit*

$$r = \begin{pmatrix} r_1 \\ \vdots \\ r_m \end{pmatrix}$$

un vecteur par blocs, formé de m vecteurs de longueur n chacun. La matrice $C = CBC(r)$ est dite circulante par blocs circulants (CBC) si elle est de la forme suivante :

$$C = \begin{pmatrix} C(r_1) & C(r_m) & \dots & C(r_2) \\ C(r_2) & C(r_1) & \dots & C(r_3) \\ \vdots & \ddots & \ddots & \vdots \\ C(r_m) & \dots & C(r_2) & C(r_1) \end{pmatrix}.$$

Comme dans le cas scalaire, on va essayer de diagonaliser les matrices CBC en utilisant les matrices de Fourrier en deux dimensions : Soit

$$F := F_m \otimes F_n. \tag{4.11}$$

Proposition 4.4.2. *La matrice C est diagonalisable par F. Plus précisement*

$$C = F^* D F,$$

avec $D = diag(Fr)$.

Démonstration. Chaque bloc, $C(r_i)$ pour $1 \leq i \leq m$, de C peut s'écrire comme $C(r_i) = F_n^* D_i F_n$, avec $D_i = \mathrm{diag}(F_n r_i)$. Donc

$$\begin{aligned} C &= \begin{pmatrix} C(r_1) & \dots & C(r_2) \\ \vdots & \ddots & \vdots \\ C(r_m) & \dots & C(r_1) \end{pmatrix} = \begin{pmatrix} F_n^* D_1 F_n & \dots & F_n^* D_2 F_n \\ \vdots & \ddots & \vdots \\ F_n^* D_m F_n & \dots & F_n^* D_1 F_n \end{pmatrix} \\ &= \begin{pmatrix} F_n^* & & \\ & \ddots & \\ & & F_n^* \end{pmatrix} \begin{pmatrix} D_1 & \dots & D_2 \\ \vdots & \ddots & \vdots \\ D_m & \dots & D_1 \end{pmatrix} \begin{pmatrix} F_n & & \\ & \ddots & \\ & & F_n \end{pmatrix} \\ &= (I_m \otimes F_n^*) D (I_m \otimes F_n), \end{aligned}$$

La matrice D est circulante par blocs diagonaux. On peut donc la transformer, en utilisant la permutation P, en une matrice diagonale par blocs circulants. En effet, Soient

$$R = \begin{pmatrix} F_n r_1 \\ \vdots \\ F_n r_m \end{pmatrix} \quad \text{et} \quad R' = PR = \begin{pmatrix} r'_1 \\ \vdots \\ r'_n \end{pmatrix},$$

les r'_i sont des vecteurs de longueur m. On peut vérifier facilement que

$$PDP^T = \begin{pmatrix} C(r'_1) & & \\ & \ddots & \\ & & C(r'_n) \end{pmatrix}.$$

Par suite

$$PDP^T = (I_n \otimes F_m^*) \begin{pmatrix} D'_1 & & \\ & \ddots & \\ & & D'_n \end{pmatrix} (I_n \otimes F_m),$$

avec, pour $1 \leq i \leq n$, $D_i = \mathrm{diag}(F_m r'_i)$. Donc

$$\begin{aligned} D &= P^T(I_n \otimes F_m^*)PP^T \begin{pmatrix} D'_1 & & \\ & \ddots & \\ & & D'_n \end{pmatrix} PP^T(I_n \otimes F_m)P \\ &= (F_n^* \otimes I_m)P^T \begin{pmatrix} D'_1 & & \\ & \ddots & \\ & & D'_n \end{pmatrix} P(F_n \otimes I_m) \\ &= (F_n^* \otimes I_m)\mathrm{diag}(P^T\tilde{R}')(F_n \otimes I_m), \end{aligned}$$

avec $\tilde{R}' = \begin{pmatrix} F_m r'_1 \\ \vdots \\ F_m r'_n \end{pmatrix}$. Ainsi $C = F^*\mathrm{diag}(P^T\tilde{R}')F$. De plus

$$P^T\tilde{R}' = P^T(I_n \otimes F_m)P(I_m \otimes F_n)R = FR$$

□

En utilisant cette proposition et la proposition (4.1.4), on peut déduire le corollaire suivant

Corollaire 4.4.3. *La multiplication de C par un vecteur et le calcul de C^{-1} coûtent $\mathcal{O}(mn \log mn)$ flops.*

On peut plonger une matrice TBT, de taille mn, dans une matrice CBC de taille $4mn$. Donc, pour multiplier T par un vecteur v, on forme un vecteur v' de longueur $4mn$, en ajoutant des zéros dans les positions corresponds à ce qu'on a ajouté à T pour former C. On multiplie C par v' puis on récupère Tv à partir de Cv'. On a donc le corollaire suivant

Corollaire 4.4.4. *La multiplication d'une matrice TBT, de taille $mn \times mn$, par un vecteur v se fait en $\mathcal{O}(mn \log mn)$ flops.*

Proposition 4.4.5. *Soit T une matrice circulante par blocs de Toeplitz et donnée en blocs par $T = C(R)$; R est le vecteur par blocs $R = (T_1^T, \ldots, T_m^T)^T$ et les T_i sont des matrices de Toeplitz de taille $n \times n$ chacun. On peut calculer T^{-1} en $\mathcal{O}(mt_n + mn \log m \log n))$ flops, t_n étant le nombre des opérations nécessaires pour inverser une matrice de Toeplitz scalaire de taille $n \times n$.*

Démonstration. Comme dans le cas scalaire, voir corollaire (2.2.10), T^{-1} est une matrice circulante par blocs, et elle est donnée par $T^{-1} = C(R')$, avec

$$R' = (F_m^* \otimes I_n)(T_1^{-T}, \ldots, T_m^{-T}).$$

Pour calculer R', on doit inverser m matrices de Toeplitz de taille $n \times n$. Et la multiplication de $(F_m^* \otimes I_n)$ par $(T_1^{-T}, \ldots, T_m^{-T})$ coûte $\mathcal{O}(mn \log m \log n)$. □

Proposition 4.4.6. *Soit T une matrice de Toeplitz triangulaire par blocs de Toeplitz bandes. On peut inverser T en $\mathcal{O}(mn \log mn)$ flops.*

Démonstration. La matrice $Z = Z_n \otimes Z_m$ engendre l'algèbre, $\mathcal{Z}$, des matrices de Toeplitz triangulaire par blocs de Toeplitz triangulaire :

$$T = \sum_{\substack{0 \le i \le n-1 \\ 0 \le j \le m-1}} t_{ij} Z_n^i \otimes Z_m^j,$$

où les t_{ij} sont les coefficients de la première colonne de T. L'algèbre $\mathcal{Z}$ est isomorphe à l'anneau des polynômes en deux variables modulo $(x^n, y^m) : T(x,y) = \sum_{\substack{0 \le i \le n-1 \\ 0 \le j \le m-1}} t_{ij} x^i \otimes y^j$. En particulier,

$$T^{-1} = \sum_{\substack{0 \le i \le n-1 \\ 0 \le j \le m-1}} t^-_{ij} Z_n^i \otimes Z_m^j \iff \left(\sum_{\substack{0 \le i \le n-1 \\ 0 \le j \le m-1}} t^-_{ij} x^i \otimes y^j \right) T(x,y) = 1 \mod (x^n, y^m)$$

□

On va donner dans ce qui suit quelques idées de résolution dans le cas général.

4.4.2 Algorithme de Wiedemann

L'algorithme de Wiedemann est un algorithme de résolution des systèmes creuses. Pour résoudre le système linéaire $Ax = b$ qui est de taille $N \times N$, cet algorithme demande N multiplications de A par un vecteur, N multiplication produits scalaires et une résolution d'un système de Toeplitz de taille $N \times N$, ce qui donne à la fin un coût total de $\mathcal{O}(k)C(A) + \mathcal{O}(N^2) + \mathcal{O}(Nlog^2N)$, où $C(A)$ est le coût de multiplication de A par un vecteur. Comme la multiplication d'une matrice TBT, de taille mn, par un vecteur se fait rapidement en $\mathcal{O}(mn)\log(mn)$, alors cet algorithme coûtera $\mathcal{O}(m^2n^2 \log(mn))$ pour un système TBT de taille mn.

Rappelons qu'un polynôme P est dit polynôme annulateur d'une matrice A si $P(A)$ est identiquement nul. Plus généralement, si b est un vecteur, un polynôme annulateur de A relativement à b est un polynôme P tel que b appartienne au noyau de $P(A)$, c'est-à-dire

$$P(A)b = \sum_{j=0}^{d} a_j A^j b = 0,$$

d étant le degré du polynôme P. Si on dispose d'un polynôme annulateur de A relativement à b, on peut alors obtenir une formule simple donnant la solution du système $Ax = b$:

$$x = -\frac{1}{a_0} \sum_{j=1}^{d} a_j A^{j-1} b. \tag{4.12}$$

Cette formule demande d multiplications de A par un vecteur, d multiplications vecteur par scalaire et $d-1$ additions de vecteur.

Afin d'obtenir un tel polynôme annulateur, Wiedemann a proposé dans [136], qui s'applique au cas des corps finis, de chercher un polynôme annulateur pour la suite $u_j = x^T A^j y$, avec x et y des vecteurs aléatoires, cela revient à chercher un polynôme non trivial $P(x) = \sum_{j=0}^{N} a_j x^j$ de degré au plus N tel que

$$\sum_{j=0}^{N} a_j u_{i+j} = 0.$$

Weidemann a évalué, dans le cas des corps finis, la probabilité pour qu'un polynôme annulateur de cette suite ne soit pas un polyôme annulateur de A. Son raisonnement s'applique au cas des corps infinis et on peut en déduire que dans ce cas, cette probabilité est nulle. Un argument topoligique simple permet aussi de déduire que cette probabilté est nulle. En effet, notons $\pi(u)$ et π_A les polynômes minimaux de A et de u respectivement. Le polynôme π_A annule clairement u, donc $\pi(u)$ divise π_A.Un argument topologique permet de montrer en fait que $\pi_u = \pi_A$, sauf en dehors de réunion finies de sous-espaces vectoriels stricts, donc en dehors d'un ensemble de mesure nulle. En effet,

1. l'ensemble
$$\{x^T \in \mathbb{K}^N; \pi((xA^n)_n) \neq \pi_A\} = \bigcup_{\substack{P|\pi_A \\ P\neq\pi_A}} \ker(P(A^T))$$
est une union finie de sous-espaces vectoriels stricts de $\mathbb{K}^N$, donc est d'intérieur vide. Si on choisit x au hasard, on peut donc supposer que $\pi((xA^n)_n) = \pi_A$.
2. pour tout $x \in \mathbb{K}^N$ tel que $\pi((x^T A^n)_n) = \pi_A$, l'ensemble
$$\{y \in \mathbb{K}^N; \pi((xA^n y)_n) \neq \pi_A\} = \bigcup_{\substack{P|\pi_A \\ P\neq\pi_A}} \ker(xP(A))$$
est une union finie de sous-espaces vectoriels stricts de $\mathbb{K}^N$, donc est d'intérieur vide. Si on choisit $y \in \mathbb{K}^N$ au hasard, on peut donc supposer que $\pi((xA^n y)_n) = \pi_A$.

On est donc réduit à trouver un plynôme annulateur de la suite des $(u_j)_{j\geq 0}$, et il suffit en particulier d'obtenir un polynôme minimal de cette suite, c'est-à-dire un polynôme annulateur monique et de degré minimal. Soit $P = x^d + \sum_{j=1}^{d-1} a_j x^j$ est un tel polynôme, supposé de degré d. Comme $P(A) = 0$ alors $A^j P(A) = 0 \, \forall j \in \mathbb{N}$, en particulier $A^j P(A) = 0$ pour $j = 0, \ldots, d-1$. Donc pour calculer les coefficients de P on utilise le système suivant :

$$\begin{cases} u_d + u_{d-1}a_{d-1} + \cdots + u_0 a_0 = 0 \\ u_{d+1} + u_d a_{d-1} + \cdots + u_1 a_0 = 0 \\ \vdots \\ u_{2d-1} + u_{2d-2}a_{d-1} + \cdots + u_{d-1} a_0 = 0 \end{cases}.$$

L'écriture matricielle de ce système est :

$$\begin{pmatrix} u_{d-1} & u_{d-2} & \ldots & a_0 \\ u_d & u_{d-1} & \ldots & u_1 \\ \vdots & \ddots & \ddots & \vdots \\ u_{2d-2} & \ldots & u_d & u_{d-1} \end{pmatrix} \begin{pmatrix} a_{d-1} \\ a_{d-2} \\ \vdots \\ a_0 \end{pmatrix} = - \begin{pmatrix} u_d \\ u_{d+1} \\ \vdots \\ u_{2d-1} \end{pmatrix}.$$

Ceci qui constitue un système de Toeplitz par rapport aux inconnues a_j. Cette matrice de Toeplitz est bien de rang plein, parce que sinon le polynôme P ne sera pas minimal. En général, d sera égal à $N = mn$, et on doit résoudre un système de Toeplitz de taill $mn \times mn$, ce qu'on sait faire en $\mathcal{O}(N \log^2 N)$ flops.

A partir de là, la resolution au moyen de la formule (4.12) sera plus intéressante qu'un élimination gaussienne seulement si la multiplication de A par un vecteur demande un nombre d'opérations petit devant N^2.

Ceci conduit à un algorithme en $\mathcal{O}(N^2 \log N)$ opérations, soit un facteur logarithmique de moins que par une méthode type Toeplitz.

4.4.3 Propriétés tensorielles des matrices de Toeplitz par blocs de Toeplitz

Une nouvelle idée pour inverser une matrice TBT est parue récemment dans [96]. L'idée est d'approcher une matrice TBT T par une matrice de petit rang de Kronecker. C'est-à-dire d'approcher T par T_r de la forme

$$T_r = \sum_{i=1}^{r} T_i^1 \otimes T_i^2, \tag{4.13}$$

avec r petit devant n et m, et pour $1 \leq i \leq r$, les T_i^1 et les T_i^2 étant de Toeplitz de taille m et n respectivement. Notons $\mathcal{T}_r$ l'ensemble de ces matrices.

On peut transformer ce problème en problème d'approximation d'une matrice par une matrice de petit rang. Donc, pour un r donné, on peut trouver la matrice $T_r \in \mathcal{T}_r$ telle que

$$\|T - T_r\| = \min_{B_r \in \mathcal{T}_r} \|T - B_r\|.$$

Pour inverser une matrice $A \in \mathcal{T}_r$ on peut envisager les itérations de Newton en posant :

$$\tilde{X}_i = 2X_{i-1} - X_{i-1}AX_{i-1}, \quad i = 0, 1, \ldots,$$

avec X_0 une approximation de A^{-1} choisi dans $\mathcal{T}_r$, et on prend X_i comme la projection de $\tilde{X}_i$ sur $\mathcal{T}_r$. Cette méthode converge quadratiquement si $\|I - AX_0\| < 1$. A chaque itération on doit faire deux multiplications matrice-matrice.

Cet article ne comporte que des essais numériques pour tenter de savoir si T_r serait une bonne approximation de T, si la méthode de Newton projetée converge, si elle converge quadratiquement, comment choisir X_0, et si le resultat, qui est un approximation de T_r^{-1} est une bonne approximation de T. Ces questions sont largement ouvertes.

Chapitre 5

Toeplitz bande par blocs Toeplitz bande

5.1 Introduction

Soit T une matrice de Toeplitz par blocs de Toeplitz à coefficients dans un corps $\mathbb{K}$. On suppose qu'elle est formée de m blocs de taille $n \times n$; de plus elle est bande par blocs, c'est à dire qu'en dehors des $2k_1 + 1$ diagonales par blocs centrales les blocs sont nuls ; les blocs eux-mêmes sont bande : en dehors des $2k_2 + 1$ diagonales centrales, les éléments des blocs sont nuls. T est donc de la forme suivante :

$$T = \begin{pmatrix} T_0 & T_{-1} & \dots & T_{-k_1} & 0 & \dots & 0 \\ T_1 & T_0 & \dots & T_{-k_1+1} & T_{-k_1} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & 0 \\ T_{k_1} & \ddots & \ddots & \ddots & \ddots & \ddots & T_{-k_1} \\ 0 & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & T_{-1} \\ 0 & \dots & 0 & T_{k_1} & \dots & T_1 & T_0 \end{pmatrix}, \tag{5.1}$$

et chaque T_j est de la forme :

$$T_j = \begin{pmatrix} T_{0,j} & T_{-1,j} & \dots & T_{-k_2,j} & 0 & \dots & 0 \\ T_{1,j} & T_{0,j} & \dots & T_{-k_2+1,j} & T_{-k_2,j} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & 0 \\ T_{k_1,j} & \ddots & \ddots & \ddots & \ddots & \ddots & T_{-k_2,j} \\ 0 & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & T_{-1,j} \\ 0 & \dots & 0 & T_{k_2,j} & \dots & T_{1,j} & T_{0,j} \end{pmatrix}. \tag{5.2}$$

Soit $N = nm$, et on va supposer que m et n sont de même ordre, c'est à dire $C^{-1}m \leq n \leq Cm$ pour une certaine constante $C > 0$. Par conséquent $m = \mathcal{O}(\sqrt{N}),\ n = \mathcal{O}(\sqrt{N})$.

Il est classique que la résolution d'un système $n \times n$ de structure bande de largeur k coûte $\mathcal{O}(k^2 n)$ opérations par la méthode de Gauss. Dans notre cas, le coût serait donc en $\mathcal{O}(N^2)$ opérations avec des méthodes directes pour matrices creuses.

Dans ce chapitre, on va essayer d'exploiter la structure *Toeplitz bande par blocs Toeplitz bande* pour donner une estimation en $\mathcal{O}(N^{3/2})$.

Remarquons tout d'abord que des statistiques expérimentales sur le nombre de conditionnement des matrices aléatoires de Toeplitz et des matrices aléatoires de Toeplitz par blocs de Toeplitz montrent des comportements dont il importe de tenir compte dans l'analyse des résultats.

La section (5.2) décrit ces statistiques ainsi que les moyens utilisés pour les obtenir. Il faut mentionner ici que les comportements dans le cas Toeplitz et dans le cas Toeplitz par blocs de Toeplitz sont qualitativement différents.

5.2 Statistiques pour des matrices bandes Toeplitz et pour des matrices bandes Toeplitz par blocs bande Toeplitz

5.2.1 État des connaissances

Un certain nombre de résultats théoriques sont connus pour des matrices aléatoires à structure bande, ou des matrices aléatoires à structure de Toeplitz. Notons en particulier un théorème de limite centrale pour un modèle de matrice bande symétrique par Anderson et Zeitouni [6], un résultat sur la distribution limite des valeurs propres pour les grandes matrices bandes symétriques ou hermitiennes par Molchanov et al. [83], un résultat de grandes déviations pour un modèle de matrices hermitiennes par Guionnet [60], des résultats sur les développements asymptotiques et les échelles d'universatité spectrale par Khorunzhy et kirsch [75]. D'autre part il y a des résultats pour les matrices de Toeplitz aléatoires , voir [81] et [14].

aucun de ces résultats ne porte sur le nombre de conditionnement de l'une ou l'autre famille de modèles de matrice, alors que c'est l'objet essentiel pour l'analyse numérique des méthodes de résolution de systèmes linéaires.

On a donc décidé d'obtenir des informations expérimentales pour des matrices aléatoires bande Toeplitz et bande Toeplitz par blocs bande Toeplitz et on a procédé comme suit.

5.2.2 Algorithmes

Notons $2k+1$ la largeur impaire de bande dans le cas de Toeplitz scalaire et $2k_1 + 1$, $2k_2 + 1$ les deux largeurs impaires de bande dans le cas par blocs. Remarquons que les diagonales non nulles occupent une zone symétrique. Plus précisément, dans le cas scalaire, les diagonales sont nulles en dehors de $\{-k, \ldots, k\}$, et dans le cas par blocs, les diagonales de blocs sont nulles en dehors de $\{-k_1, \ldots, k_1\}$ et les diagonales dans les blocs sont nulles en dehors de $\{-k_2, \ldots, k_2\}$.

On a généré des coefficients pseudo-aléatoires gaussiens en nombre approprié, c'est à dire k dans le cas scalaire et k_1k_2 dans le cas par blocs, ce qui permet de décrire une matrice T en structure creuse.

On a effectué un décomposition LU creuse, par l'algorithme "superLU", de T et sa transposée, ce qui a permis de calculer la plus petite valeur singulière par la méthode de la puissance inverse. D'autre part on a obtenu la plus grande valeur singulière par la méthode de la puissance.

On a constaté la convergence rapide de la méthode de la puissance inverse, et la convergence très lente de la méthode de la puissance. Ce qui n'est pas entièrement surprenant.

Le rapport entre la plus grande et la plus petite des valeurs singulières fournit le nombre de conditionnement $\kappa(T) = \|T\|_2\|T^{-1}\|_2$, la norme $\|.\|_2$ étant la norme spectrale, c'est à dire la racine carrée de la plus grande valeur propre de T^*T.

5.2.3 Les résultats dans le cas scalaire

On se reportera au figure (5.1) pour voir des histogrammes du logarithme en base 10 du nombre de conditionnement des matrices de Toeplitz de même taille et des largeurs de bande différentes.

On remarque que ces histogrammes dépendent peu de la largeur de bande.

5.2.4 Les résultats dans le cas par blocs

Dans ce cas on remarque que la statistique sur les nombres de conditionnements dépend des largeurs de bande. On constate qu'avec l'accroissement d'au moins une des largeurs de bande, l'histogramme devient de plus en plus étroit, et la moyenne du logarithme du nombre de conditionnement décroît. Ceci montre que les matrices TBT bande par blocs bande de petites largeurs de bande sont plus mal conditionnées que celles de largeurs de bande plus grandes.

On se reportera aux figures (5.2), (5.3), (5.4) et (5.5).

Dans la suite, on décrira trois algorithmes rapides de résolution d'une matrice bande en commençant tout d'abord par le cas scalaire, puis en les généralisant au cas par blocs.

5.3 Cas scalaire

Les idées de deux premiers algorithmes proviennent de Bini et Pan [9], et les auteurs mentionnent un problème d'instabilité pour le premier algorithme.

5.3.1 Transformation en matrice circulante plus matrice de petit rang

Rappelons, tout d'abord, la Formule de Sherman-Morrison-Woodbury :

Théoreme 5.3.1. *Soient $A \in \mathbb{K}^{n\times n}$, G, $H \in \mathbb{K}^{n\times k}$. Si $I_k + H^TA^{-1}G$ n'est pas singulière, alors*

$$(A + GH^T)^{-1} = A^{-1} - A^{-1}G(I_k + H^TA^{-1}G)^{-1}H^TA^{-1}. \tag{5.3}$$

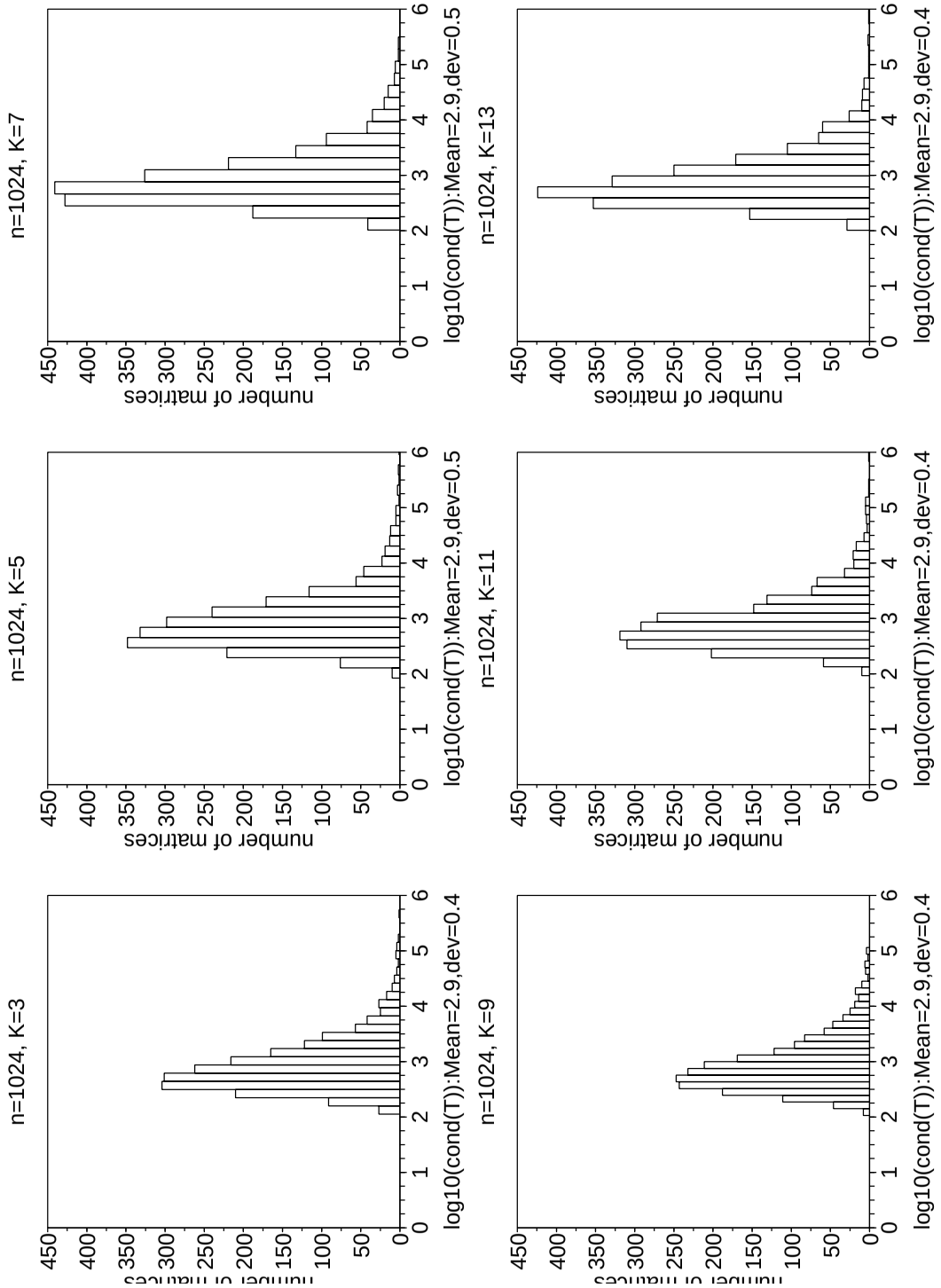


FIG. 5.1 – Les matrices ici sont de Toeplitz bande de taille 1024×1024, la largeur de bande, $2k+1$, varie entre 3 et 13. On remarque que le nombre de conditionnement ne varie pas avec la largeur de la bande.

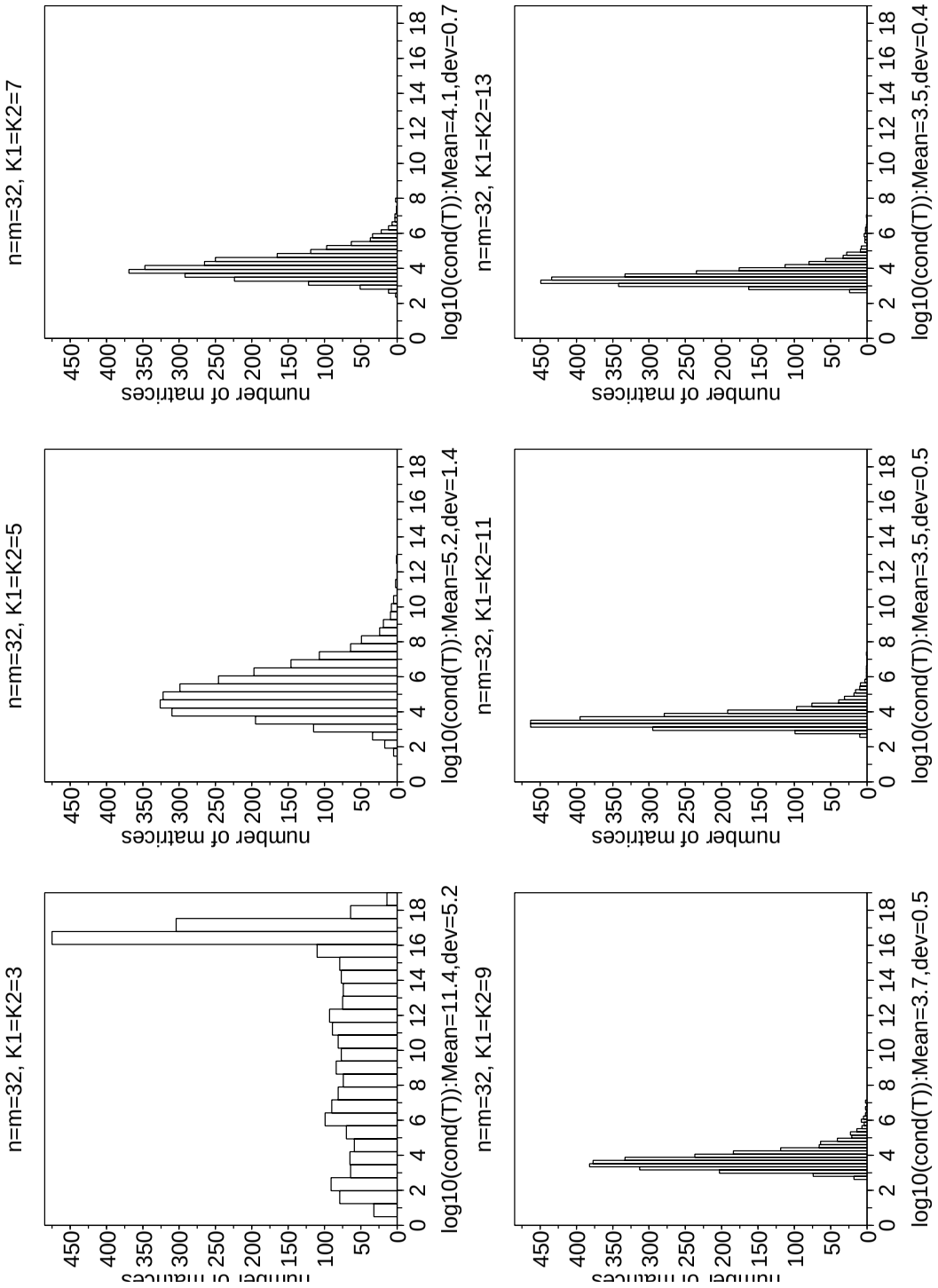


FIG. 5.2 – On considère des matrices TBT de taille $32^2 \times 32^2$ et de largeur de bande $K_1 = 2k_1 + 1 = 2k_2 + 1$, allant de 3 à 13. On a fait 2500 essaies pour chaque cas.

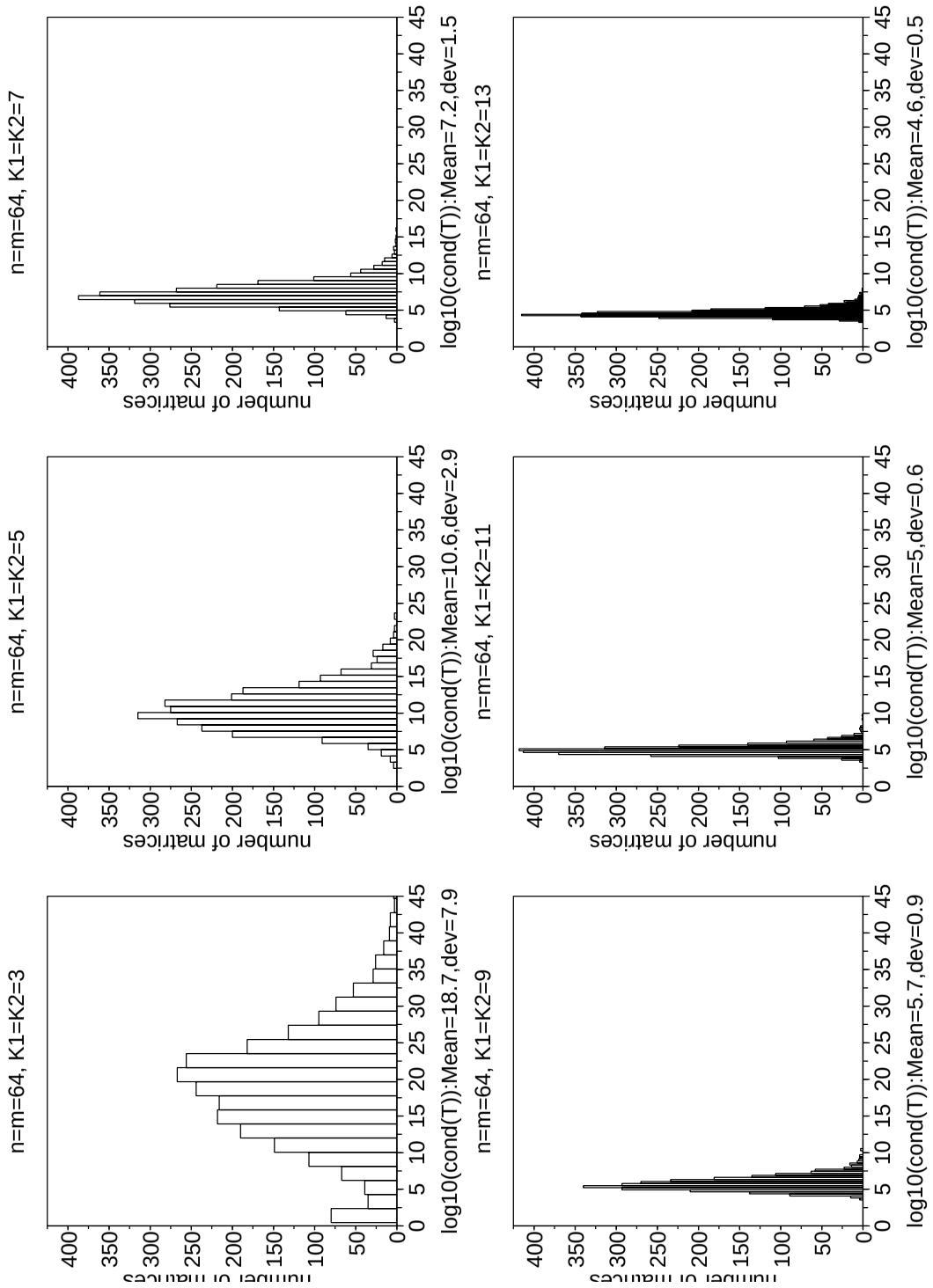


FIG. 5.3 – Les matrices ici sont de taille $64^2 \times 64^2$. La largeur des bandes et le nombre des essaies sont comme pour Fig. 5.2

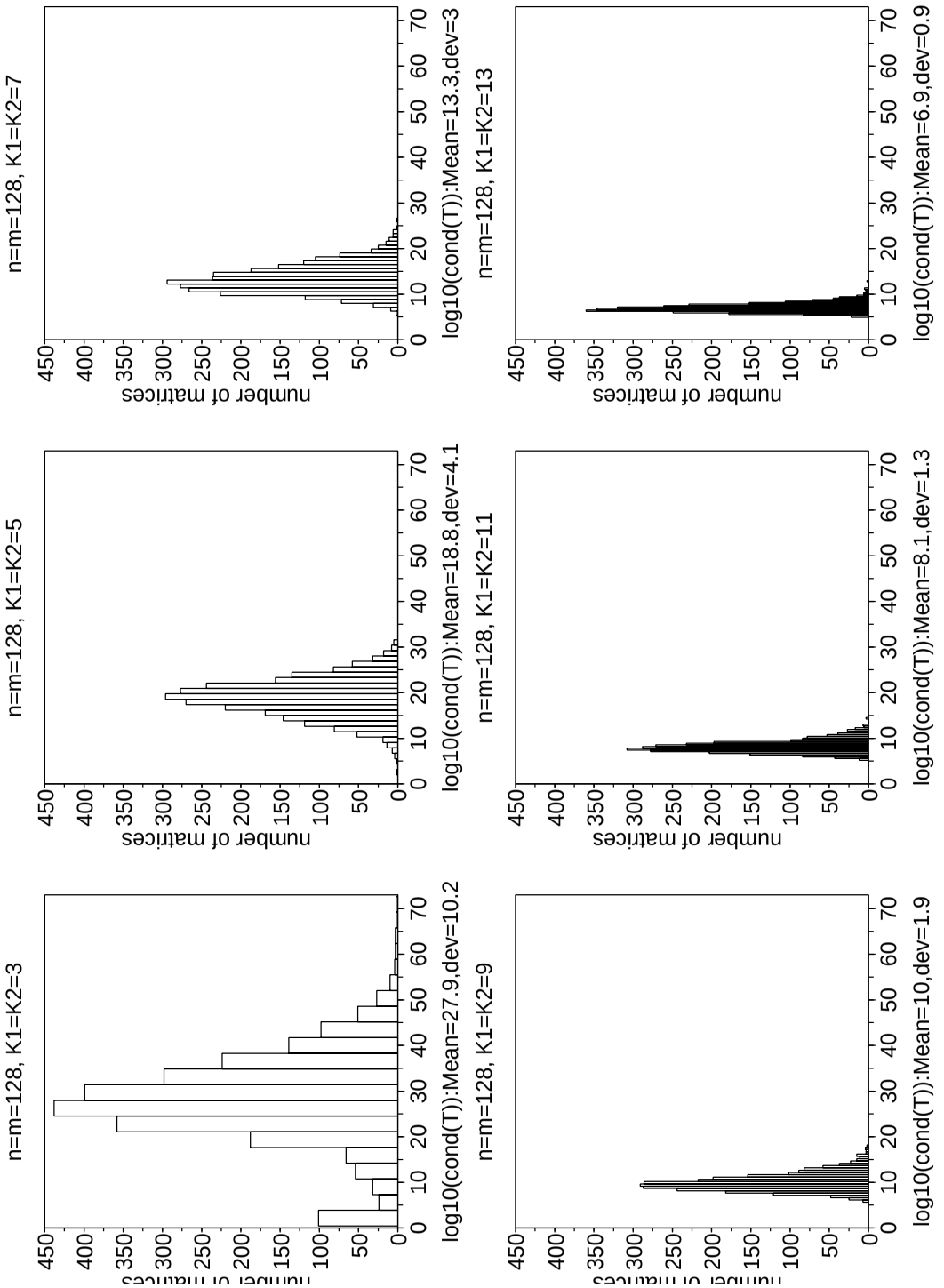


FIG. 5.4 – Matrices de Taille $128^2 \times 128^2$.

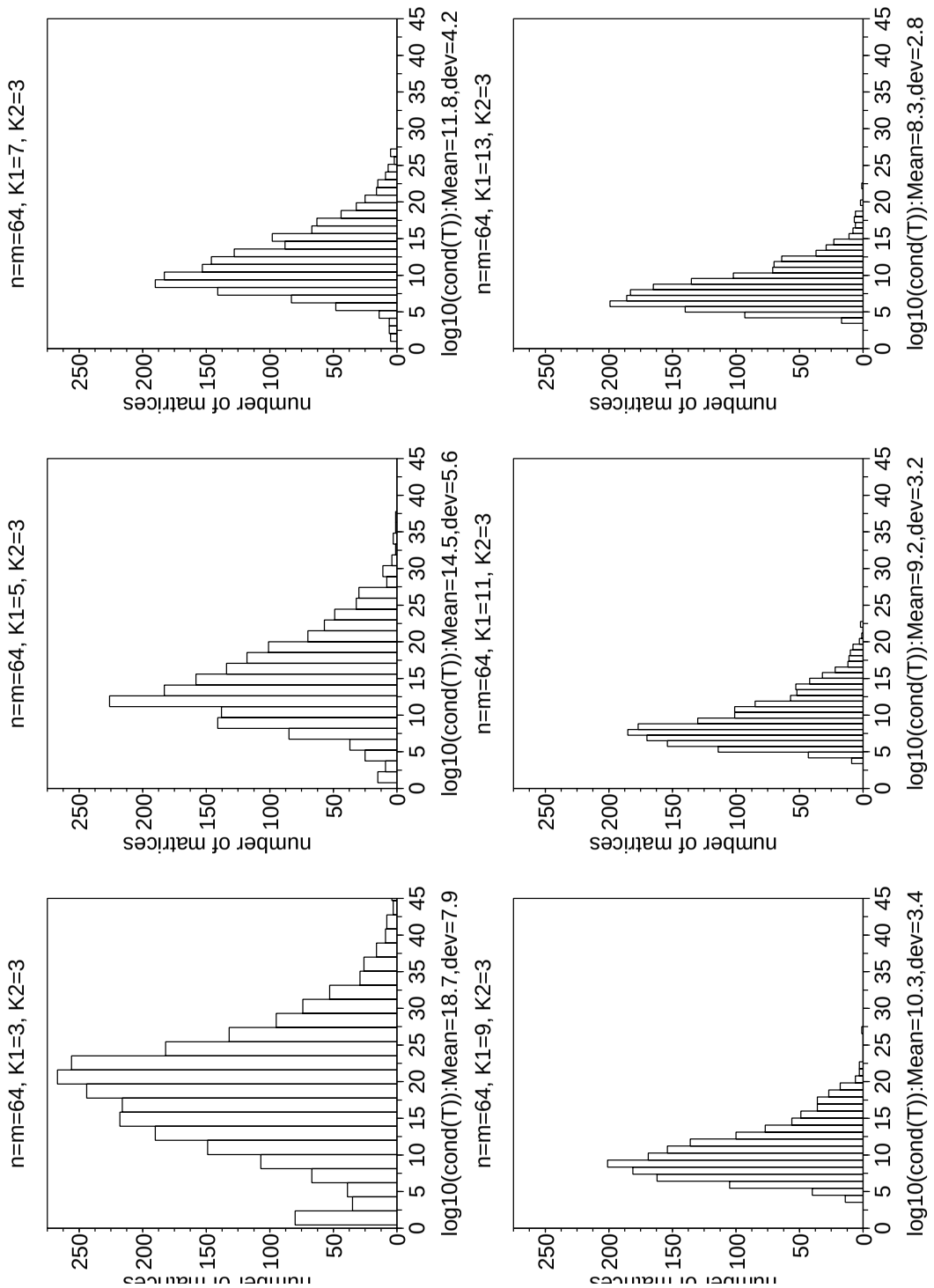


FIG. 5.5 – Les matrices sont de taille $64^2 \times 64^2$. Ici $k_1 = 1$ est fixe, $K_2 = 2k_2 + 1$ varie entre 3 et 13. Cet histogramme met en évidence la dépendance entre largeur des bandes et nombre de conditionnement quand la largeur d'une seule bande est variable.

Démonstration. Une matrice par blocs peut se factoriser comme suit :

$$L = \begin{pmatrix} M & N \\ P & Q \end{pmatrix} = \begin{pmatrix} I & 0 \\ PM^{-1} & I \end{pmatrix} \begin{pmatrix} M & 0 \\ 0 & S \end{pmatrix} \begin{pmatrix} I & M^{-1}N \\ 0 & I \end{pmatrix},$$

avec S le complément de Schur donné par $S = Q - PM^{-1}N$. Bien entendu cette factorisation n'a de sens que si M est inversible. Par symétrie, on a également

$$L = \begin{pmatrix} M & N \\ P & Q \end{pmatrix} = \begin{pmatrix} I & NQ^{-1} \\ 0 & I \end{pmatrix} \begin{pmatrix} \tilde{S} & 0 \\ 0 & Q \end{pmatrix} \begin{pmatrix} I & 0 \\ Q^{-1}P & I \end{pmatrix},$$

avec $\tilde{S} = M - NQ^{-1}P$. En inversant les deux termes de l'égalité et en identifiant les deux expressions du bloc de la première ligne et de la première colonne on tire l'identité :

$$M^{-1} + M^{-1}NS^{-1}PM^{-1} = \tilde{S}^{-1}$$

qu'on applique à

$$L = \begin{pmatrix} A & G \\ H^T & -I_k \end{pmatrix}.$$

□

Proposition 5.3.2. *Soit $A \in \mathbb{K}^{n\times n}$ une matrice de Toeplitz bande, de largeur de bande $2k+1$. Alors on peut décomposer A sous la forme $C + R$ avec C une matrice circulante et R une matrice de rang au plus $2k$.*

Démonstration. La décomposition suivante est immédiate :

$$A = \begin{pmatrix} a_0 & a_{-1} & \dots & a_{-k} & 0 & \dots & 0 \\ a_1 & a_0 & \dots & a_{-k+1} & a_{-k} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & 0 \\ a_k & \ddots & \ddots & \ddots & \ddots & \ddots & a_{-k} \\ 0 & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & a_{-1} \\ 0 & \dots & 0 & a_k & \dots & a_1 & a_0 \end{pmatrix} \tag{5.4}$$

$$= \begin{pmatrix} a_0 & \dots & a_{-k} & 0 & a_k & \dots & a_1 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ a_k & \ddots & \ddots & \ddots & \ddots & \ddots & a_k \\ 0 & \ddots & \ddots & \ddots & \ddots & \ddots & 0 \\ a_{-k} & \ddots & \ddots & \ddots & \ddots & \ddots & a_{-k} \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ a_{-1} & \dots & a_{-k} & 0 & a_k & \dots & a_0 \end{pmatrix} - \begin{pmatrix} 0 & \dots & 0 & a_k & \dots & a_1 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & \ddots & a_k \\ a_{-k} & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ a_{-1} & \dots & a_{-k} & 0 & \dots & 0 \end{pmatrix}$$

$$= C((a_0 \dots a_k\, 0 \dots 0\, a_{-k} \dots a_{-1})^T) + R = C + R.$$

□

Corollaire 5.3.3. *Supposons que A vérifie les hypothèses de la proposition 5.3.2. Alors, il existe une méthode directe pour résoudre le système $Ax = b$ en $\mathcal{O}(n \log n) + \mathcal{O}(nk \log k) + \mathcal{O}(k^3)$ opérations.*

Démonstration. En écrivant $R = GH^T$, avec H, $G \in \mathbb{K}^{n \times r}$ $(r = 2k)$, et en utilisant la formule de Sherman-Morrison-Woodbury sur le système $(C + R)x = b$ on obtient

$$x = C^{-1}b - C^{-1}G(I_r + H^T C^{-1} G)^{-1} H^T C^{-1} b.$$

Remarquons tout d'abord que G est creuse avec $k(k+1)$ éléments non nuls :

$$G = \begin{pmatrix} 0_{k\times k} & g \\ 0_{(n-2k)\times k} & 0_{(n-2k)\times k} \\ f & 0_k \end{pmatrix} \text{ et } H = \begin{pmatrix} I_k & 0_{k\times k} \\ 0_{(n-2k)\times k} & 0_{(n-2k)\times k} \\ 0_{k\times k} & I_k \end{pmatrix},$$

avec $f = L(a_{-k} \dots a_{-1})^T$ et $g = U(a_k \dots a_1)^T$, et $L(v)$ (resp. $U(v)$ la matrice de Toeplitz triangulaire inférieure (resp. triangulaire supérieure) de première colonne (resp. première ligne) égale à v. Ainsi la multiplication d'une matrice de taille $n \times n$ par G coûte $\mathcal{O}(nk^2)$ opérations (la multiplication d'une matrice creuse qui contient p éléments non nuls par un vecteur coûte $2p$ opérations). On peut même faire mieux, en multipliant G par une matrice $n \times n$ en $\mathcal{O}(nk \log k)$ comme f et g sont de Toeplitz.

Par suite x est obtenu en faisant :
- $v_1 = C^{-1}b$: $\mathcal{O}(n \log n)$ opérations,
- $v_2 = H^T v_1$: 0 opérations, parcequ'il n'y a pas que des 0 et des 1 dans H,
- $C^{-1}v_1$, avec $v_1 = G(I_r + H^T C^{-1} G)^{-1} H^T C^{-1} b$.
- $H^T C^{-1} G$: $\mathcal{O}(n \log n) + \mathcal{O}(nk \log k)$ opérations (calculer C^{-1} en $\mathcal{O}(n \log n)$ opérations puis la multiplier par G en $\mathcal{O}(nk \log k)$ opérations),
- $v_3 = (I_r + H^T C^{-1} G)^{-1} v_2$: $\mathcal{O}(k^3)$ opérations,
- $v_4 = G v_3$: $\mathcal{O}(k \log k)$ opérations,
- $v_5 = C^{-1} v_4$: $\mathcal{O}(n \log n)$ opérations.

□

Remarque 5.3.4. *On a $\mathcal{O}(nk \log k)$ est plus 'grand' que $\mathcal{O}(n \log n)$ si $k = \mathcal{O}(n^\alpha)$ pour un $\alpha \in [0, 1[$.*

Instabilité de l'algorithme

Dans leur livre, *Polynomial and matrix computation*, Victor Pan et Dario Bini, énoncent que cet algorithme rencontre des problèmes de stabilité. Les problèmes peuvent être dûs à l'instabilité propre du système linéaire. En effet, on pourra constater sur les figures que la comparaison entre nombre de conditionnement et erreur est normale. Pour se faire, nous avons généré des matrices de Toeplitz (et de Toeplitz bande par blocs Toeplitz bandes. Nous avons testé l'algorithme sur les deux types de matrices) bande pseudo aléatoires uniformes. Nous avons également généré des vecteurs x aléatoires et nous avons calculé l'erreur entre x et $\tilde{x}$ qui est la *solution numérique*, via notre algorithme, de $T\tilde{x} = b$, b étant égal à Tx. Cette comparaison semble montrer que l'instabilité ne vient pas de l'algorithme.

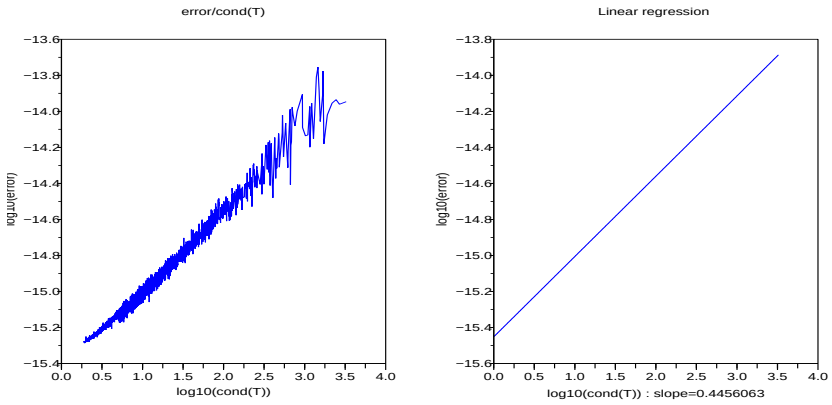


FIG. 5.6 – Les matrices ici sont de Toeplitz bande de taille 400×400 et la largeur de la bande $2k+1 = 5$. Pour 3000 essaies, on trace le logarithme à base 10 de l'erreur dû à notre algorithme par rapport au logarithme à base 10 du nombre de conditionnement. On trouve que la pente de la droite de régression est plus petit que 0.5 !

5.3.2 Plongement dans une matrice engendrée par $Z + Z^T$

On va décrire maintenant un deuxième algorithme de résolution d'une matrice de Toeplitz bande. Dans cette section on travaille avec des matrices carrées de taille n. Soit τ_n (τ s'il y a pas de confusion) l'algèbre engendrée par $W = Z + Z^T$. Les matrices de Toeplitz dans cette section sont symétriques.

Proposition 5.3.5. *Soit $A \in \tau$, alors ses coefficients vérifient*

$$\begin{aligned} a_{i-1,j} + a_{i+1,j} &= a_{i,j+1} + a_{i,j-1} \\ a_{0,j} = a_{n+1,j} &= a_{i,0} = a_{i,n+1} = 0 \end{aligned} \tag{5.5}$$

Démonstration. Si A est dans τ, elle est de la forme $A = \sum_{i=0}^{n-1} \alpha_i W^i$. Or $W^{k+1} = W^k.W = W.W^k$ et si M une matrice alors $(MW)_{ij} = M_{i,j+1} + M_{i,j-1}$ et $(WM)_{i,j} = M_{i-1,j} + M_{i,j+1}$. Donc

$$(W^{k+1})_{ij} = (W^k)_{i,j-1} + (W^k)_{i,j+1} = (W^k)_{i-1,j} + (W^k)_{i+1,j}. \tag{5.6}$$

Montrons par récurrence sur k que chaque W^k vérifie la propriété (5.5) : W la vérifie. Supposons que W^k est telle que

$$(W^k)_{i,j-1} + (W^k)_{i,j+1} = (W^k)_{i-1,j} + (W^k)_{i+1,j},$$

D'après (5.6) :

$$(W^k)_{i,j-1} + (W^k)_{i,j+1} = (W^k)_{i-1,j-1} + (W^k)_{i+1,j-1} + (W^k)_{i-1,j+1} + (W^k)_{i+1,j+1},$$

et

$$(W^k)_{i-1,j} + (W^k)_{i+1,j} = (W^k)_{i-1,j-1} + (W^k)_{i-1,j+1} + (W^k)_{i+1,j-1} + (W^k)_{i+1,j+1},$$

parsuite $(W^k)_{i,j-1} + (W^k)_{i,j+1} = (W^k)_{i-1,j} + (W^k)_{i+1,j}$. □

Corollaire 5.3.6. *De là nous déduisons que l'algèbre τ_n peut être identifiée à*

$$B = \{B_k \in \tau; B_k e_1 = e_k,\ 1 \leq k \leq n\}.$$

Pour $A \in \tau$,

$$A = \sum_{k=1}^{n} a_k B_k,$$

avec $(a_1, \ldots, a_n)^T$ la première ligne de A.

Démonstration. Pour construire une matrice de l'algèbre τ il suffit de connaître sa première colonne. Or, comme B_k, $1 \leq k \leq n$, est la matrice de τ_n telle que sa première colonne est e_k, et comme la première colonne de A vaut $\sum_{k=1}^{n} a_k e_k$, alors $A = \sum_{k=1}^{n} a_k B_k$. □

Proposition 5.3.7. *Soit Δ l'espace vectoriel des matrices de Toeplitz $n \times n$, de largeur de bande $2k+1$. Soit dans $\tau_{n+2(k+1)}$, le sous espace vectoriel engendré par les matrices B_i telles que $B_i e_1 = e_i - e_{i-2}$, avec $i = 3 \ldots k+1$, ainsi que $B_1 = I, B_2 = W$. Soit $E = \{k+1, \ldots, n-k-1\}$. Alors les matrices $\tilde{B}_i = ((B_i)_{lp})_{l,p \in E}$ forment une base de Δ.*

Démonstration. En utilisant la technique donnée dans la proposition précédente pour calculer un élément dans τ, on remarque que B_i est de la forme :

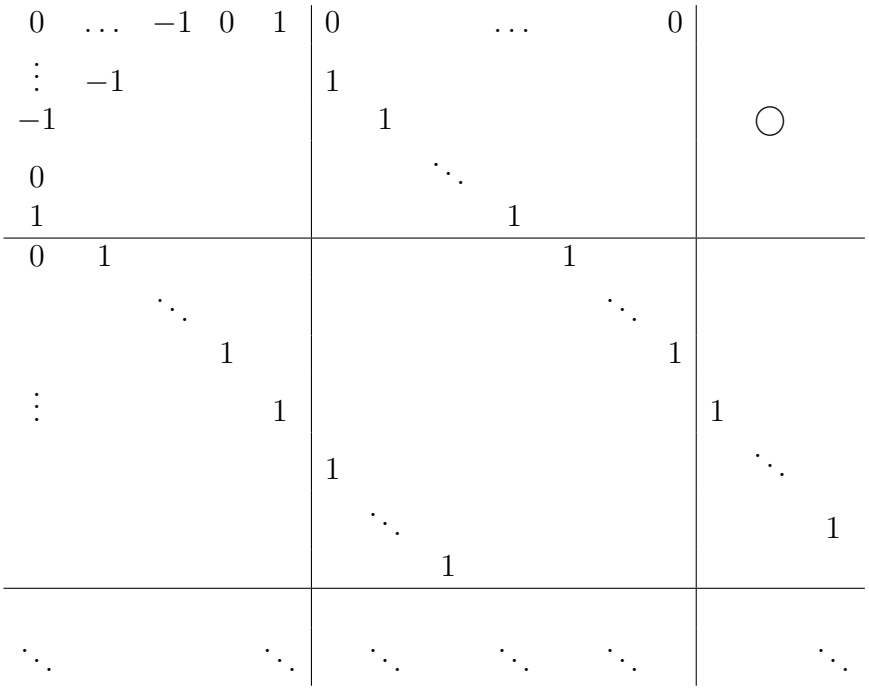


FIG. 5.7 – La matrice B_i comporte un centre, c'est à dire l'intersection des lignes et des colonnes indexées par $E = \{k+1, \ldots, n-k-1\}$, et une périphérique qui est son complémentaire

et $\tilde{B}_i$ est la matrice qui contient des 1 sur les diagonales qui commencent par les éléments $(1, i)$ et $(i, 1)$ respectivement. Par suite les $\tilde{B}_i$ génèrent Δ □

Soit $T \in \Delta$ telle que $T = \sum_{i=1}^{k+1} a_i \tilde{B}_i$. On peut donc la plonger dans une matrice $M \in \tau$ avec $M = \sum_{i=1}^{k+1} a_i B_i$ de taille $n + 2(k+1)$. M a donc la forme suivante :

$$M = \begin{pmatrix} M_{11} & M_{12} & M_{13} \\ M_{12}^T & T & M_{23} \\ M_{13}^T & M_{23}^T & M_{33} \end{pmatrix},$$

avec M_{11}, M_{13} et M_{33} de taille $(k+1) \times (k+1)$, M_{12} et M_{23}^T de taille $(k+1) \times n$. On cherche à résoudre le système $Tx = b$. Étudions le système suivant :

$$\begin{pmatrix} M_{11} & M_{12} & M_{13} \\ M_{12}^T & T & M_{23} \\ M_{13}^T & M_{23}^T & M_{33} \end{pmatrix} \begin{pmatrix} 0 \\ x \\ 0 \end{pmatrix} = \begin{pmatrix} b_1 \\ b \\ b_3 \end{pmatrix}.$$

Dans ce système, les inconnues sont x, b_1 et b_3 et on a $Tx = b$. Pour trouver b_1 et b_3 on procède comme suit : soit

$$M^{-1} = \begin{pmatrix} \mu_{11} & \mu_{12} & \mu_{13} \\ \mu_{12}^T & \mu_{22} & \mu_{23} \\ \mu_{13}^T & \mu_{23}^T & \mu_{33} \end{pmatrix}.$$

Les vecteurs b_1 et b_3 vérifient :

$$\begin{pmatrix} \mu_{11} & \mu_{12} & \mu_{13} \\ \mu_{12}^T & \mu_{22} & \mu_{23} \\ \mu_{13}^T & \mu_{23}^T & \mu_{11} \end{pmatrix} \begin{pmatrix} b_1 \\ b \\ b_3 \end{pmatrix} = \begin{pmatrix} 0 \\ x \\ 0 \end{pmatrix},$$

et par suite résolvent le système

$$\begin{cases} \mu_{11} b_1 + \mu_{13} b_3 = -\mu_{12} b, \\ \mu_{13}^T b_1 + \mu_{33} b_3 = -\mu_{23}^T b. \end{cases}$$

Ce système, de taille $2(k+1) \times 2(k+1)$, donne b_1 et b_3 en $\mathcal{O}(k^3)$ opérations. Pour construire ce système on a besoin aussi de calculer μ_{11}, μ_{12}, μ_{13} et μ_{23}, puis de calculer $\mu_{12} b$ et $\mu_{23}^T b$. comme μ_{12} et μ_{23}^T sont de taille $(k+1) \times n$, ce calcul va coûter $\mathcal{O}(nk)$ opérations. Le calcul de μ_{11}, μ_{12}, μ_{13}, μ_{23} coûte $\mathcal{O}(n \log^2(n)) + \mathcal{O}(k^2 n)$ opérations (voir corollaire ci-dessus).

Proposition 5.3.8. *Soit $M \in \tau_n$. On peut résoudre le système $Mx = b$ en $\mathcal{O}(n \log^2 n)$ opérations.*

Démonstration. $M \in \tau$ Comme M est dans τ_n, elle est de la forme

$$M = \sum_{i=1}^{n-1} m_i W^i.$$

Or les valeurs propres et vecteurs propres de W sont donnés par :

$$\lambda_k = 2 \cos \frac{k\pi}{n+1}, \quad v_k = \left(\sin \frac{jk\pi}{n+1} \right)_{1 \le j \le n}, \quad k = 1 \dots n,$$

comme on peut le vérifier simplement. Soit donc s la matrice de la transformation de Fourier en sinus :

$$s = \begin{pmatrix} \sin\frac{\pi}{n+1} & \dots & \sin\frac{n\pi}{n+1} \\ \vdots & \ddots & \vdots \\ \sin\frac{n\pi}{n+1} & \dots & \sin\frac{n^2\pi}{n+1} \end{pmatrix} = \left(\sin\frac{ij\pi}{n+1}\right)_{1\le i,j\le n}.$$

Classiquement, $s^{-1} = 2s/(n+1)$. Si on pose $S = s\sqrt{2/(n+1)}$, alors $S^{-1} = S$ et $W = SDS$ avec D la matrice diagonale des valeurs propres. Nous en déduisons la diagonalisation de M :

$$M = S\left(\sum_{i=0}^{n-1} m_i D^i\right) S.$$

Le calcul de la somme des $m_i D^i$ se fait en $\mathcal{O}(n\log^2 n)$ (évaluation d'un polynôme (avec coefficients m_i) aux n valeurs propres λ_k). □

Corollaire 5.3.9. *Soit $M \in \tau_{n+2(k+1)}$. Pour calculer la périphérie de taille $k+1$ de M^{-1} on a besoin de $\mathcal{O}((n+k)\log^2(n+k)) + \mathcal{O}(nk^2)$ opérations.*

Démonstration. Notons $K = k+1$. La matrice M est symétrique, elle est aussi symétrique par rapport à l'antidiagonale. Donc, pour calculer les K premières lignes, les K premières colonnes, les K dernières lignes et les K dernières colonnes de M^{-1} (la périphérie de M^{-1}) il suffira de calculer les K premières colonnes. On a donc besoin de résoudre K systèmes linéaires avec la matrice M. D'après la proposition précédente ce calcul coûtera $\mathcal{O}(K(n+2K)\log^2(n+2K))$ opérations.

On peut construire la périphérie d'une autre manière. En effet, comme $M^{-1} \in \tau$, on peut calculer sa première colonne en $\mathcal{O}((n+2K)\log(n+2K))$ opérations, et construire les autres colonnes demandées en utilisant la technique de la proposition (5.3.5), ce qui demande $\mathcal{O}(nk^2)$ opérations. □

Corollaire 5.3.10. *La résolution de $Tx = b$ coûte $\mathcal{O}(n\log^2 n) + \mathcal{O}(k^3)$ opérations.*

5.3.3 Plogement dans une matrice circulante

L'idée de cet algorithme est de plonger la matrice de Toeplitz bande dans une matrice circulante à la place de la plonger dans une matrice de l'algèbre τ. Dans cette section, les matrices de toeplitz bandes ne sont pas nécessairement symétriques.

Proposition 5.3.11. *Soit $A \in \mathbb{K}^{n\times n}$ une matrice de Toeplitz bande, de largeur de bande $2k+1$. Alors on peut plonger A dans une matrice circulante de taille $(n+k)\times(n+k)$.*

Proposition 5.3.12. *Soit A une matrice de Toeplitz bande donnée comme dans (5.4). On peut donc plonger A dans la matrice circulante $C = C(r)$ de première colonne r donné par*

$$r = (\underbrace{a_0, \dots, a_k, 0, \dots, 0}_{n}, a_{-k}, \dots, a_{-1})^T.$$

En effet,

$$C = \left(\begin{array}{cccccccc|ccc} a_0 & \dots & a_{-k+1} & a_{-k} & 0 & \dots & 0 & a_k & \dots & a_1 \\ \vdots & \ddots & & & \ddots & & & & \ddots & \vdots \\ a_{k-1} & & \ddots & & & \ddots & & & & a_k \\ a_k & & & \ddots & & & a_{-k} & & & \\ 0 & \ddots & & & \ddots & & & a_{-k} & & \\ \vdots & & \ddots & & & \ddots & \vdots & \vdots & \ddots & \\ 0 & & & a_k & & \dots & a_0 & a_{-1} & \dots & a_{-k} \\ \hline a_{-k} & & & & a_k & \dots & a_1 & a_0 & \dots & a_{-k+1} \\ \vdots & \ddots & & & & \ddots & \vdots & \vdots & \ddots & \vdots \\ a_{-1} & \dots & a_{-k} & & & & a_k & a_{k-1} & \dots & a_0 \end{array}\right) \tag{5.7}$$

$$= \left(\begin{array}{c|c} A & B \\ \hline D & T \end{array}\right)$$

avec T une matrice de Toeplitz de taille $k \times k$, D et B^T sont de Toeplitz, creuses, de taille $k \times n$.

Comme dans l'algorithme précédent, on réduit la résolution du système $Ax = b$ à la résolution du système

$$C \begin{pmatrix} x \\ 0 \end{pmatrix} = \begin{pmatrix} b \\ c \end{pmatrix}$$

avec c un vecteur inconnu de longueur k.

En écrivant

$$C^{-1} = \begin{pmatrix} \gamma_{11} & \gamma_{12} \\ \gamma_{21} & \gamma_{22} \end{pmatrix},$$

avec γ_{11} de taille $n \times n$ et γ_{22} de taille $k \times k$, alors le vecteur c résolve le système de Toeplitz, de taille $k \times k$, suivant :

$$\gamma_{22} c = -\gamma_{21} b.$$

On peut donc calculer c en $\mathcal{O}(k \log^2 k)$ flops. Le vecteur $(x\, 0)^T$ est la solution d'un système circulant. Donc, une fois calculer c, $(x\, 0)^T$ sera calculé en $\mathcal{O}((n+k)\log(n+k))$ flops. Cet algorithme est un peu plus rapide que les deux premiers parceque le calcul de c peut se faire en $\mathcal{O}(k \log^2 k)$ flops.

Proposition 5.3.13. *La résolution du système $Ax = b$ coûte $\mathcal{O}(n \log n) + \mathcal{O}(kn) + \mathcal{O}(k \log^2 k)$ flops.*

Démonstration. Le calcul de $\gamma_{21} b$ demande $\mathcal{O}(kn)$ flops. Parsuite le calcul de c demande $\mathcal{O}(k \log^2 k)$, et à la fin la résolution du système $C \begin{pmatrix} x \\ 0 \end{pmatrix} = \begin{pmatrix} b \\ c \end{pmatrix}$ se fait en $\mathcal{O}(n \log n)$ flops. □

5.4 Cas par blocs

Le cas par blocs est une généralisation directe du cas scalaire mais où y rencontre quelques complications, et les comptes d'opérations sont différents.

5.4.1 Transformation en matrice circulante plus matrice de petit rang

Soit T une matrice comme dans (5.1) et (5.2).

Proposition 5.4.1. *On peut décomposer T en $T = C + R$, avec C une matrice circulante par blocs circulants et R de rang au plus $2(k_1 n + k_2 m)$, et au plus $\mathcal{O}(k_1^2 k_2 n + k_2^2 k_1 m)$ éléments non nuls.*

Démonstration. Écrivons $T_i = C_i + \tilde{R}_i$, avec $-k_1 \leq i \leq k_1$, où C_i est une matrice circulante et $\tilde{R}_i$ une matrice de rang au plus $2k_2$. Nous pouvons alors décomposer T comme suit :

$$
\begin{aligned}
T = &\begin{pmatrix}
C_0 & \dots & C_{-k_1} & 0 & C_{k_1} & \dots & C_1 \\
\vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\
C_{k_1} & \ddots & \ddots & \ddots & \ddots & \ddots & C_{k_1} \\
0 & \ddots & \ddots & \ddots & \ddots & \ddots & 0 \\
C_{-k_1} & \ddots & \ddots & \ddots & \ddots & \ddots & C_{-k_1} \\
\vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\
C_{-1} & \dots & C_{-k_1} & 0 & C_{k_1} & \dots & C_0
\end{pmatrix} - \\
&\begin{pmatrix}
0 & \dots & 0 & C_{k_1} & \dots & C_1 \\
\vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\
0 & \ddots & \ddots & \ddots & \ddots & C_{k_1} \\
C_{-k_1} & \ddots & \ddots & \ddots & \ddots & 0 \\
\vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\
C_{-1} & \dots & C_{-k_1} & 0 & \dots & 0
\end{pmatrix} - \begin{pmatrix}
\tilde{R}_0 & \dots & \tilde{R}_{-k_1} & 0 & \dots & 0 \\
\vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\
\tilde{R}_{k_1} & \ddots & \ddots & \ddots & \ddots & 0 \\
0 & \ddots & \ddots & \ddots & \ddots & \tilde{R}_{-k_1} \\
\vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\
0 & \dots & 0 & \tilde{R}_{k_1} & \dots & \tilde{R}_0
\end{pmatrix} \\
&= C + R_1 + R_2 = C + R.
\end{aligned}
$$

Nous remarquons que R_1 est au plus de rang $2k_1 n$ et R_2 est de rang $2k_2 m$.

Dans R_1 il y a $2 \times k_1(k_1+1)/2$ blocs non nuls, et dans chaque bloc C_i il y a $n + 2(n-1) + 2(n-2) + \cdots + 2(n-k_2) + (k_2^2 + k_2) = 2k_2 n + n$ éléments non nuls. Ainsi R_1 contient $(k_1^2 + k_1)(2k_2 n + n) = \mathcal{O}(k_1^2 k_2 n)$ élément non nuls ; la situation est analogue pour R_2. □

En écrivant $R = GH^T$, avec G, $H \in \mathbb{K}^{N \times r}$ $(r = 2(k_1 n + k_2 m))$, et en appliquant la formule de Sherman-Morrison-Woodbury sur le système $(C + R)x = b$ on obtient

$$x = C^{-1}b - C^{-1}G(I_r + H^T C^{-1} G)^{-1} H^T C^{-1} b.$$

Proposition 5.4.2. *G est creuse avec $\mathcal{O}(k_1^2k_2n)+\mathcal{O}(k_2^2k_1m)$ éléments non nuls, et H ne comprend que des 0 et des 1.*

Démonstration. Les matrices G et H se décomposent en

$$G = (G_1\, G_2) \text{ et } H^T = \begin{pmatrix} H_1^T \\ H_2^T \end{pmatrix}$$

avec

$$G_1 = \begin{pmatrix} 0_{nk_1} & E \\ \vdots & 0 \\ 0 & \vdots \\ F & 0_{nk_1} \end{pmatrix},$$

pour décrire la matrice G_2, notons

$$E_l = \{(l-1)n+1, \dots, (l-1)n+k_2\} \cup \{ln-k_2+1, \dots, ln\}$$

et

$$E = \cup_{l=1}^{m} E_l$$

alors

$$G_2 = ((R_2)_{ij})_{\substack{1\le i\le mn \\ j\in E}},$$

$$H_1^T = \begin{pmatrix} I_{nk_1} & 0 & \dots & 0_{nk_1} \\ 0_{nk_1} & \dots & 0 & I_{nk_1} \end{pmatrix},$$

et

$$H_2^T = \left(\begin{array}{cc|c|cc} I_{k_2} & 0_{k_2\times(n-k_2)} & \dots & I_{k_2} & 0_{k_2\times(n-k_2)} \\ 0_{k_2\times(n-k_2)} & I_{k_2} & & 0_{k_2\times(n-k_2)} & I_{k_2} \end{array}\right)$$

chaque bloc dans H_2 est de taille $2k_2 \times m$.

Donc, en utilisant le compte fait dans la proposition précédente on voit que G contient $(k_1^2+k_1)(2k_2n+n)+(k_2^2+k_2)(2k_1m+m) = \mathcal{O}(k_1^2k_2n)+\mathcal{O}(k_2^2k_1m)$. □

Corollaire 5.4.3. *La multiplication de G par un vecteur coûte $\mathcal{O}(k_1^2k_2n)+\mathcal{O}(k_2^2k_1m)$ opérations.*

Corollaire 5.4.4. *En supposant que k_1 et k_2 sont petit devant m et n respectivement alors la résolution du système $Tx = b$ donné en (5.1) et (5.2) coûte $\mathcal{O}(N^{3/2})$ opérations.*

Démonstration. Soit $N = nm$, $K = k_1^2k_2n + k_2^2k_1m$, et $r = 2(k_1n + k_2m)$.

On a $x = C^{-1}b - C^{-1}G(I_r + H^TC^{-1}G)^{-1}H^TC^{-1}b$, donc x est obtenu en faisant :

- $v_1 = C^{-1}b$: $\mathcal{O}(N\log N)$.
- $v_2 = H^Tv_1$: 0 opération, car il n'y a que des 0 et des 1 dans H.
- $H^TC^{-1}G$: $\mathcal{O}(N\log N)+\mathcal{O}(NK) = \mathcal{O}(N\log N)+\mathcal{O}(N^{3/2})$, le calcul de C^{-1} demande $\mathcal{O}(N\log N)$ opérations et la multiplication par G demande $\mathcal{O}(NK)$ opérations car G est creuse.
- $v_3 = (I_r + H^TC^{-1}G)^{-1}v_2$: $\mathcal{O}(r^3) = \mathcal{O}(N^{3/2})$, résolution classique.
- $v_4 = Gv_3$: $\mathcal{O}(K) = \mathcal{O}(N^{1/2})$, multiplication de G par un vecteur, donc proportionnel au nombre d'éléments non nuls dans G.
- $v_5 = C^{-1}v_4$: $\mathcal{O}(N\log N)$.

□

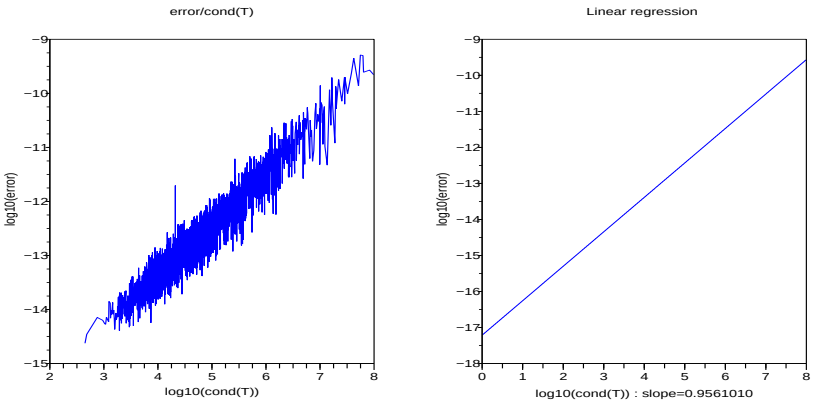


FIG. 5.8 – Les matrices sont de Toeplitz bande par blocs Toeplitz bande de taille $32^2 \times 32^2$ de largeur de bande $K_1 = 2k_1 + 1 = K_2 = 2k_2 + 1 = 5$. Pour 3000 essaies, on trace le logarithme à base 10 de l'erreur dû à notre algorithme par rapport au logarithme à base 10 du nombre de conditionnement. On trouve que la pente de la droite de régression est à peu près égale 1.

Stabilité

Comme dans le cas scalaire, on a appliqué cet algorithme sur des matrices de Toeplitz bande par blocs Toeplitz bandes pseudo aléatoires pour obtenir les erreurs entre la solution de $Tx = b$ calculée par notre algorithme et la solution exacte. En traçant ces erreurs par rapport aux nombres de conditionnement, on a obtenu une ligne de régression de pente $\simeq 1$. Ce qui prouve qu'il y a pas un problème de stabilité pour cet algorithme. Mais on remarque que la pente est significativement augmenté de $\simeq 0.5$ pour le cas scalaire à $\simeq 1$ pour le cas par blocs!!!

5.4.2 Plongement dans une matrice engendrée par $Z + Z^T$

Dans cette section T est une matrice de Toeplitz, par blocs de Toeplitz symétrique par blocs, et les blocs sont symétriques, bande par blocs, et les blocs sont aussi bandes. Les entiers m, n, k_1 et k_2 sont les dimensions utilisées dans les sections précédents. Pour résoudre le système $Tx = b$, on va essayer de plonger la matrice T dans une matrice de l'algèbre $\tau_{m,n}$ (ou τ s'il y a pas de confusion) engendrée par $W = (Z_m + Z_m^T) \otimes (Z_n + Z_n^T)$. Une matrice $M \in \tau$ est donc de la forme

$$M = \sum_{\substack{0 \le i \le m-1 \\ 0 \le j \le n-1}} \alpha_{ij} (Z_m + Z_m^T)^i \otimes (Z_n + Z_n^T)^j = \sum \alpha_{i,j} W^{(i,j)}.$$

Si M est de plus bande par blocs bandes, les bornes supérieures respectives de i et j dans la sommation sont k_1 et k_2. On va essayer de procéder comme dans le cas scalaire pour avoir un algorithme rapide.

Proposition 5.4.5. *Soit M dans l'algèbre τ. On peut résoudre le système $Mx = b$ en $\mathcal{O}(nm \log^2 mn)$ opérations.*

Démonstration. Notons s_k la matrice de la transformation en sinus de dimension k, $S_k = s_k\sqrt{2/(k+1)}$ et D_k la matrice diagonale $\mathrm{diag}(\lambda_1, \ldots, \lambda_k)$. Comme $M \in \tau$ alors $M = \sum \alpha_{ij} W^{(i,j)}$ avec $W := W_m \otimes W_n$ et $W^{(i,j)} := (Z_m + Z_m^T)^i \otimes (Z_n + Z_n^T)^j$. Or $W = W_m \otimes W_n = (S_m D_m S_m) \otimes (S_n D_n S_n) = (S_m \otimes S_n)(D_m \otimes D_n)(S_m \otimes S_n) =: SDS$. Donc M admet la diagonalisation :

$$M = SES \text{ avec } E = \sum_{\substack{0 \le i \le m-1 \\ 0 \le j \le n-1}} \alpha_{ij} D_m^i \otimes D_n^j.$$

E peut être calculé en $\mathcal{O}(mn \log^2 mn)$ opérations : il s'agit d'évaluer un polynôme à deux variable en mn points. Comme M^{-1} admet la décomposition $M^{-1} = SE^{-1}S$, la multiplication de M^{-1} par un vecteur coûte 2 transformées rapide en sinus et une multiplication par une matrice diagonale, le tout en $\mathcal{O}(mn \log mn)$ opérations. □

Soit T une matrice bande Toeplitz symétrique par blocs bande Toeplitz symétrique, comme en (5.1). De même technique que pour le cas scalaire, nous la plongeons dans une matrice $M \in \tau$, en la plongeant tout d'abord par blocs, puis par blocs et après en plongeant chaque bloc.

La matrice M obtenue est une matrice de taille $m + 2(k_1 + 1)$ par blocs, et chaque bloc est de taille $n + 2(k_2 + 1)$; donc M est de taille $(m + 2(k_1 + 1))(n + 2(k_2 + 1)) \simeq mn + 2mk_1 + 2nk_2 + 4k_1k_2$. Elle a la forme suivante :

$M =$

$$\left(\begin{array}{c|c|c}
M_{11} & M_{12} & M_{13} \\
\hline
M_{21} & \begin{array}{|ccc|}\hline \times & \times & \times \\ \times & T_0 & \times \\ \times & \times & \times \\ \hline \end{array} \quad \cdots \quad \begin{array}{|ccc|}\hline \times & \times & \times \\ \times & T_{n-1} & \times \\ \times & \times & \times \\ \hline \end{array} \\
 & \vdots \qquad\qquad \ddots \qquad\qquad \vdots \\
 & \begin{array}{|ccc|}\hline \times & \times & \times \\ \times & T_{n-1} & \times \\ \times & \times & \times \\ \hline \end{array} \quad \cdots \quad \begin{array}{|ccc|}\hline \times & \times & \times \\ \times & T_0 & \times \\ \times & \times & \times \\ \hline \end{array} & M_{23} \\
\hline
M_{31} & M_{32} & M_{33}
\end{array}\right), \tag{5.8}$$

avec $M_{11}, M_{13}, M_{31}, M_{33}$ de taille $(k_1+1)(n+2(k_2+1)) \times (k_1+1)(n+2(k_2+1))$ et M_{12}, M_{32} sont de taille $(k_1 + 1)(n + 2(k_2 + 1)) \times m(n + 2(k_2 + 1))$.

On cherche à résoudre $Tx = b$ avec

$$x = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}, \quad b = \begin{pmatrix} b_1 \\ \vdots \\ b_n \end{pmatrix}, \quad \text{les } x_i \text{ et les } b_i \text{ étant des vecteurs de taille } n.$$

Comme dans le cas scalaire, on va compléter le vecteur x par de nouvelles composantes nulles, dont les numéros de ligne sont les numéros des nouvelles ligne de M par rapport à T. Il faudra donc ajouter des lignes à b, ce qui créera de nouvelles inconnues :

$$\tilde{x} = \begin{pmatrix} 0_k \\ \bar{x} \\ 0_k \end{pmatrix} \text{ et } \tilde{b} = \begin{pmatrix} \hat{b}_1 \\ \bar{b} \\ \hat{b}_2 \end{pmatrix},$$

avec $k = (k_1 + 1)(n + (k_2 + 1))$ et

$$\bar{x} = \left(\begin{array}{c} 0_{\bar{k_2}} \\ x_1 \\ 0_{\bar{k_2}} \\ \hline \vdots \\ \hline 0_{\bar{k_2}} \\ x_m \\ 0_{\bar{k_2}} \end{array}\right) \text{ et } \bar{b} = \left(\begin{array}{c} \hat{b}_{11} \\ b_1 \\ \hat{b}_{13} \\ \hline \vdots \\ \hline \hat{b}_{m1} \\ b_m \\ \hat{b}_{m3} \end{array}\right),$$

avec $\bar{k_2} = (k_2 + 1)$, les $\hat{b}_{i1}$ et les $\hat{b}_{i3}$ de taille $\bar{k_2}$. Soit

$$\hat{b} = \left(\begin{array}{c} \hat{b}_1 \\ \hat{b}_{11} \\ b_1 \\ \hat{b}_{13} \\ \vdots \\ \hat{b}_{m1} \\ b_m \\ \hat{b}m3 \\ \hat{b}_3 \end{array}\right)$$

Comme dans le cas scalaire, écrivons $\tilde{x} = M^{-1}\tilde{b}$, et en ne gardant dans ce système que les lignes nulles dans $\tilde{x}$ et que les nouvelles inconnues, nous obtenons un système par rapport à $\hat{b}$, dont la matrice est de taille $\simeq nk_1 + mk_2$. Pour décrire ce système on va écrire M^{-1} d'une façon équivalente à l'écriture de M donnée en (5.8), elle sera donc donnée comme suit :

$$M^{-1} = \left(\begin{array}{c|c|c} \mu_{11} & \mu_{12} & \mu_{13} \\ \hline \mu_{21} & \left(\begin{array}{c|c|c} \mu_{11}^{ij} & \mu_{12}^{ij} & \mu_{13}^{ij} \\ \hline \mu_{21}^{ij} & \mu^{ij} & \mu_{23}^{ij} \\ \hline \mu_{31}^{ij} & \mu_{32}^{ij} & \mu_{33}^{ij} \end{array}\right)_{i,j=1\ldots m} & \mu_{23} \\ \hline \mu_{31} & \mu_{32} & M_{33} \end{array}\right),$$

on découpe aussi μ_{12} (pareil pour μ_{32}) de la façon suivante :

$$\mu_{12} = \left(\mu_{12,11} \mid \mu_{12,1} \mid \mu_{12,13} \mid \dots \mid \mu_{12,m1} \mid \mu_{12,m} \mid \mu_{12,m3} \right).$$

Le système à résoudre sera donné par :

$$\begin{cases} \mu_{11}\hat{b}_1 + \sum_{i=1}^{m}(\mu_{12,i1}\hat{b}_{i1} + \mu_{12,i3}\hat{b}_{i3}) + \mu_{13}\hat{b}_3 = -\sum_{i=1}^{m}\mu_{12,i}b_i \\ \mu_{12,j1}^T\hat{b}_1 + \sum_{i=1}^{m}(\mu_{11}^{ij}\hat{b}_{i1} + \mu_{13}^{ij}\hat{b}_{i3}) + \mu_{23,j1}\hat{b}_3 = -\sum_{i=1}^{m}\mu_{12}^{ij}b_i \ j = 1\dots m \\ \mu_{12,j3}^T\hat{b}_1 + \sum_{i=1}^{m}(\mu_{31}^{ij}\hat{b}_{i1} + \mu_{33}^{ij}\hat{b}_{i3}) + \mu_{23,j3}\hat{b}_3 = -\sum_{i=1}^{m}\mu_{32}^{ij}b_i \ j = 1\dots m \\ \mu_{31}\hat{b}_1 + \sum_{i=1}^{m}(\mu_{32,i1}\hat{b}_{i1} + \mu_{32,i3}\hat{b}_{i3}) + \mu_{33}\hat{b}_3 = -\sum_{i=1}^{m}\mu_{12,i}b_i \end{cases}$$

Pour former le deuxième membre de ce système il faut $\mathcal{O}(k_1(n+k_2).m(n+2k_2)) + \mathcal{O}(2m^2.k_2.n) \simeq \mathcal{O}(N^{3/2}) + \mathcal{O}(N^{3/2})$ opérations : 2 multiplications d'une matrice de taille $(k_1+1)(n+(k_2+1)) \times m(n+2(k_2+1))$ par un vecteur et $2m^2$ multiplications matrice-vecteur avec des matrices de taille $(k_2+1)n$.

Proposition 5.4.6. *En supposant que k_1 et k_2 sont petit devant m et n respectivement alors le calcul de la périphérie (en deux dimension) de M^{-1} coûte $\mathcal{O}((n+k_1)(m+k_2)\log((n+k_1)(m+k_2))) + \mathcal{O}(N^{3/2})$ opérations.*

Démonstration. Même démonstration que pour le cas scalaire □

Finalement, la résolution du système initiale requiert $\mathcal{O}(N^{3/2})$ opérations pour être résolu.

5.4.3 Plogement dans une matrice circulante par blocs circulants

Soit T une matrice bande Toeplitz par blocs bande Toeplitz comme en (5.1). On s'intéresse au problème $Tx = b$ avec

$$x = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}, \quad b = \begin{pmatrix} b_1 \\ \vdots \\ b_n \end{pmatrix}, \quad \text{les } x_i \text{ et les } b_i \text{ étant des vecteurs de taille } n.$$

En suivant les téchniques du cas scalaire, on peut plonger T dans une matrice circulante par blocs circulants, C, de taille $(m+k_1)(n+k_2) \times (m+k_1)(n+k_2)$. En effet, on plonge tout d'abord chaque bloc T_i, pour $-k_1 \leq i \leq k_1$, dans une matrice circulante de taille $(n+k_2) \times (n+k_2)$. Par suite on plonge par bloc la matrice obtenue, qui est une matrice bande Toeplitz par blocs, dans une matrice circulante par blocs. La matrice finale est bien une matrice circulante par blocs circulants de taille $(m+k_1)(n+k_2) \times (m+k_1)(n+k_2)$.

On suit les téchniques de la section précédente : on pose le problème

$$C\tilde{x} = \tilde{b}$$

avec

$$\tilde{x} = \begin{pmatrix} \bar{x} \\ 0_k \end{pmatrix} \quad \text{et} \quad \tilde{b} = \begin{pmatrix} \bar{b} \\ c \end{pmatrix}$$

avec $k = k_1(n + k_2)$, c un vecteur inconnu de longueur k et

$$\bar{x} = \begin{pmatrix} x_1 \\ 0_{k_2} \\ \hline \vdots \\ \hline x_m \\ 0_{k_2} \end{pmatrix} \quad \text{et} \quad \bar{b} = \begin{pmatrix} b_1 \\ c_1 \\ \hline \vdots \\ \hline b_m \\ c_m \end{pmatrix}$$

où chaque c_i, $1 \leq i \leq m$, est un vecteur inconnu de longueur k_2.

Comme dans le cas scalaire, pour trouver le vecteur inconnu

$$\zeta = \begin{pmatrix} c_1 \\ \vdots \\ c_m \\ c \end{pmatrix}$$

on inverse a matrice C et on extraire un système llinéaire,$M\zeta = g$, dont la solution est ζ. Ce système sera de taille $K \times K$, avec $K = mk_2 + nk_1 + k_1k_2$. On peut remarquer quelque structure dans M, mais elle n'est pas TBT (dans le cas scalaire, on a obtenu un sytème de Toeplitz). Le calcul de g demande une multiplication d'une matrice de taille $k_1(n+k_2)\times m(n+k_2)$ par un vecteur, et m^2 multiplications des matrices de taille $k_2 \times n$ par des vecteurs. Ceci demande en total, pour calculer g, $\mathcal{O}(k_1mn^2+2k_1k_2mn+k_1k_2^2m+k_2m^2n)$ flops, qui est équivalent à $\mathcal{O}(N^{3/2})$ flops. La résolution du système $M\zeta = g$ demande $\mathcal{O}(K^3)$ flops, ce qui est aussi équivalent à $\mathcal{O}(N^{3/2})$ flops.

A la fin, la résolution du système $C\tilde{x} = \tilde{b}$ se fait en $\mathcal{O}((m+k_1)(n+k_2)\log(m+k_1)(n+k_2))$ flops, qui est équivalent à $\mathcal{O}(N \log N)$.

Cet algorithme est donc d'orde $N^{3/2}$ flops.

Chapitre 6

Matrices de Toeplitz et de Toeplitz par blocs de Toeplitz et leurs relation avec les syzygies des polynômes

6.1 Introduction

Les matrices de Toeplitz et de Toeplitz par blocs de Toeplitz (TBT), et plus généralement les matrices structurées, sont très liées aux polynômes, voir [44, 9]. On a vu, dans (2.4.1), que la multiplication d'une matrice de Toeplitz par un vecteur peut être déduite de la multiplication de deux polynômes d'une seule variable. Dans [88], on trouve comment lier entre l'inverse d'une matrice de Hankel ou de Toeplitz et le Bezoutien des polynômes associés à leurs générateurs. On peut aussi déduire la multiplication d'une matrice de Toeplitz par blocs de Toeplitz par un vecteur de la multiplication de deux polynômes en deux variables. Voir [88] pour une description plus générale de la relation entre les matrices TBT et polynômes.

Les difficultés, discutées dans le chapitre (3), nous ont poussés à exploiter ces relations entre matrices TBT et polynômes en deux variables, pour essayer de trouver des solveurs super rapides. On n'est pas arrivé à ce but, mais on a, quand même, donné des nouvelles relations entre solution d'une matrice TBT et syzygies des polynômes en deux variables et on a donné une nouvelle méthode super rapide pour résoudre les systèmes de Toeplitz scalaires.

On a étudié, tout d'abord, la résolution d'un système de Toeplitz scalaire, $T\,u = g$, d'un nouveau point de vue, en donnant une relation entre la solution d'un tel problème et les syzygies des polynômes d'une seule variable. On donne une connexion explicite entre les générateurs d'une matrice de Toeplitz et les générateurs du module des syzygies correspondant. On démontre que ce module est engendré par deux éléments de degré n et que la solution de $T\,u = g$ peut être interprétée comme le reste d'un vecteur, associé à g, par ces deux générateurs.

Cette approche s'étend naturellement aux problèmes en plusieurs variables. On démontre, tout d'abord, comment on peut étendre la notion des générateurs en une matrice TBT T, puis on décrit la structure des générateurs du module des syzygies correspondants

aux générateurs de T, et comment on peut déduire la solution du problème $T\,u = g$ à partir de ces générateurs. Ce qui donne un nouveau point de vue pour résoudre les système de Toeplitz par blocs de Toeplitz. On donne aussi une nouvelle preuve élémentaire d'un résultat connu sur les modules de syzygies, c'est que le module des syzygies de k polynômes, non nuls, en deux variables est libre de rang $k-1$.

On commencera, dans la section suivante, par l'étude du cas scalaire. Puis, on traite le cas de Toeplitz par blocs de Toeplitz dans la section 3.

Dans ce chapitre, on utilisera les notations suivantes.

- $R = \mathbb{K}[x]$, l'ensemble des polynômes en x et à coefficients dans le corps $\mathbb{K}$..
- Pour $n \in \mathbb{N}$, on dénote par $\mathbb{K}[x]_n$ le k-espace vectoriel des polynômes de degré $\leq n$.
- $L = \mathbb{K}[x, x^{-1}]$, l'ensemble des polynôme de Laurent pour la variable x et à coefficients dans $\mathbb{K}$.
- Pour un polynôme $p = \sum_{i=-m}^{n} p_i\, x^i \in L$, on dénote par p_+ la somme des termes des exposants positifs : $p_+ = \sum_{i=0}^{n} p_i\, x^i$, et par p_-, la somme des termes avec des exposant strictement négatifs : $p_- = \sum_{i=-m}^{-1} p_i\, x^i$. On a bien sûr $p = p_+ + p_-$.
- Pour $n \in \mathbb{N}$, on dénote par $\mathfrak{U}_n = \{\omega; \omega^n = 1\}$ l'ensemble des racines $n^{\text{ème}}$ de l'unité.
- Pour un vecteur $u = (u_0, \ldots, u_{k-1})^T \in \mathbb{K}^k$, on dénote par $u(x)$ le polynôme de degré $k-1$ donné par $u(x) = \sum_{i=0}^{k-1} u_i x^i$. Inversement, si $v(x) = \sum_{i=0}^{k-1} v_i x^i$ est un polynôme de degré $k-1$, on dénote par v le vecteur de longueur k des coefficients de $v(x)$.

6.2 Cas scalaire

On commence par le cas d'une seule variable et par le problème suivant :

Problème 6.2.1. *Soit $T = (t_{i-j})_{i,j=0}^{n-1} \in \mathbb{K}^{n\times n}$ une matrice de Toeplitz de taille $n \times n$ et $g = (g_0, \ldots, g_{n-1}) \in \mathbb{K}^n$, trouver $u = (u_0, \ldots, u_{n-1}) \in \mathbb{K}^n$ tel que*

$$T\,u = g. \tag{6.1}$$

Définition 6.2.2. *Soit $E = \{1, \ldots, x^{n-1}\}$ l'ensemble des monômes de degré $< n$, et soit Π_E la projection de L sur l'espace vectoriel engendré par E, parallèlement à $\langle x^n, x^{n+1}, \ldots\rangle$.*

A partir de la matrice T, des vecteurs g et u, on va définir les polynômes suivants :

Définition 6.2.3. *Soient*

- $T(x) = \displaystyle\sum_{i=-n+1}^{n-1} t_i x^i$,
- $\tilde{T}(x) = \displaystyle\sum_{i=0}^{2n-1} \tilde{t}_i x^i$ *avec* $\tilde{t}_i = \begin{cases} t_i & \text{if } i < n \\ t_{i-2n} & \text{if } i \geq n \end{cases}$,
- $u(x) = \displaystyle\sum_{i=0}^{n-1} u_i x^i$, $g(x) = \displaystyle\sum_{i=0}^{n-1} g_i x^i$.

Remarquons que $T(x)$ est un polynôme de Laurent et que $\tilde{T}(x)$ est la transformation de ce polynôme de Laurent en un vrai polynôme en remplaçant les exposants négatifs par des exposants de degré $> n$. On a donc la proposition suivante :

Proposition 6.2.4. *$\tilde{T} = T_+ + x^{2n} T_-$ et $T(w) = \tilde{T}(w)$ pour $w \in \mathfrak{U}_{2n}$.*

Démonstration. On peut déduire directement, à partir de la définition de $T(x)$ et $\tilde{T}(x)$, que $\tilde{T} = T_+ + x^{2n} T_-$. De plus, comme $w^{2n} = 1$ et comme $T(x) = T_+(x) + x^{2n} T_-(x)$, alors $\tilde{T}(w) = T_+(w) + T_-(w) = T(w)$ □

D'après la proposition 2.1.2 de [88] (voir aussi l'équation (2.8) du chapitre (2)), on a cette relation entre le problème (6.1) et les polynômes :

Proposition 6.2.5.

$$T\,u = g \Leftrightarrow \Pi_E(T(x)u(x)) = g(x).$$

Or, comme $\Pi_E(T(x)u(x))$ n'est que le polynôme $T(x)u(x)$ dont on enlève les termes de degré négatif et de degré $\geq n$, on peut déduire la proposition, dans le cas général, suivante. Notons, tout d'abord : pour tout polynôme $u \in \mathbb{K}[x]$ de degré d, on écrit : $u(x) = \underline{u}(x) + x^n \overline{u}(x)$ avec $\deg(\underline{u}) \leq n-1$ et $\deg(\overline{u}) \leq d-n$ si $d \geq n$ et $\overline{u} = 0$ sinon.

Proposition 6.2.6. *On a*

$$T(x)\,u(x) = \Pi_E(T(x)\underline{u}(x)) + \Pi_E(T(x)x^n\overline{u}(x)) + x^{-n+1}A(x) + x^n B(x), \tag{6.2}$$

où $A(x)$ et $B(x)$ sont deux polynômes de degré $\leq n-2$ et $\leq \max(n-2, d-1)$ respectivement.

Démonstration.

$$\begin{aligned} T(x)\,u(x) &= T(x)\underline{u}(x) + T(x)x^n\overline{u}(x) \\ &= \Pi_E(T(x)\underline{u}(x)) + \Pi_E(T(x)x^n\overline{u}(x)) \\ &\quad +(\alpha_{-n+1}x^{-n+1} + \cdots + \alpha_{-1}x^{-1}) \\ &\quad +(\alpha_n x^n + \cdots + \alpha_{n+m}x^{n+m}) \\ &= \Pi_E(T(x)\underline{u}(x)) + \Pi_E(T(x)x^n\overline{u}(x)) \\ &\quad +x^{-n+1}A(x) + x^n B(x), \end{aligned} \tag{6.3}$$

avec $m = \max(n-2, d-1)$,

$$\begin{aligned} A(x) &= \alpha_{-n+1} + \cdots + \alpha_{-1}x^{n-2}, \\ B(x) &= \alpha_n + \cdots + \alpha_{n+m}x^m. \end{aligned} \tag{6.4}$$

□

Corollaire 6.2.7. *Si $u(x)$ est de degré $\leq n-1$ alors*

$$T(x)\,u(x) = \Pi_E(T(x)u(x)) + x^{-n}A(x) + x^n B(x),$$

avec $A(x)$ et $B(x)$ sont deux polynômes de degré $\leq n-1$. (Plus précisement, $\deg B(x) \leq n-2$ et $A(x)$ n'a pas un terme constant).

6.2.1 Syzygies et matrices de Toeplitz

On considère ici, un autre problème, lié à la géométrie algébrique effective :

Problème 6.2.8. *Soient a, b, c, trois polynômes dans R de degré $< l, < m, < n$ respectivement. Trouver trois polynômes $p, q, r \in R$ de degré $< \nu - l, < \nu - m, < \nu - n$, tels que :*

$$a(x)\,p(x) + b(x)\,q(x) + c(x)\,r(x) = 0. \tag{6.5}$$

Définition 6.2.9. *L'ensemble des triplets de polynômes, $(p, q, r) \in \mathbb{K}[x]^3$, qui vérifient l'équation* (6.5)*, s'appelle l'ensemble des syzygies de (a, b, c). On dénote cet ensemble par $\mathcal{L}(a, b, c)$.*

Donc, l'ensemble des solutions du problème (6.2.8) est l'ensemble $\mathcal{L}(a, b, c) \cap \mathbb{K}[x]_{\nu-l-1} \times \mathbb{K}[x]_{\nu-m-1} \times \mathbb{K}[x]_{\nu-n-1}$.

Définition 6.2.10. *Soit $d(x)$ un polynôme dans R. On dénote par $\mathcal{L}(a, b, c; d)$ l'ensemble des $(p, q, r) \in \mathbb{K}[x]^3$ tel que*

$$a(x)\,p(x) + b(x)\,q(x) + c(x)\,r(x) = d(x).$$

Théoreme 6.2.11. *$\mathcal{L}(a, b, c)$ forme un $\mathbb{K}[x]$-module de $\mathbb{K}[x]^3$. De plus, pour n'importe quel vecteur, non nul, des polynômes $(a, b, c) \in \mathbb{K}[x]^3$, l'ensemble $\mathcal{L}(a, b, c)$ est un $\mathbb{K}[x]$-module libre de rang* 2*.*

Démonstration. Par le théorème de Hilbert, l'idéal I engendré par (a, b, c) admet une résolution libre de longueur au plus 1, elle est donc de la forme :

$$0 \to \mathbb{K}[x]^p \to \mathbb{K}[x]^3 \to \mathbb{K}[x] \to \mathbb{K}[x]/I \to 0.$$

Comme $I \neq 0$, pour des raisons de dimension, on a obligatoirement $p - 3 + 1 = 0$. Voir [29] chapitre 6, pour plus d'information sur les résolutions libres et le théorème de Hilbert sur les syzygies. □

Définition 6.2.12. *Pour un vecteur de polynômes $p = (p_1, \ldots, p_k) \in R^k$, on définit* $\deg(p) = \max(\deg(p_1), \ldots, \deg(p_k))$.

Définition 6.2.13. *Supposons que* $\deg(p, q, r) \leq \deg(p', q', r')$*. Une μ-base de $\mathcal{L}(a, b, c)$ est une base $\{(p, q, r), (p', q', r')\}$ de $\mathcal{L}(a, b, c)$, avec* $\deg(p, q, r) = \mu$.

On démontre la proposition suivante, qui donne une relation entre les degrés des deux éléments d'un générateur de $\mathcal{L}(a, b, c)$.

Proposition 6.2.14. *Si (p, q, r) et (p', q', r'), de degrés μ_1 et μ_2 respectivement, forment un système générateur de $\mathcal{L}(a, b, c)$, alors $d = max(\deg(a), \deg(b), \deg(c)) = \mu_1 + \mu_2$.*

Démonstration. On a

$$0 \to \mathbb{K}[x]_{\nu-d-\mu_1} \oplus \mathbb{K}[x]_{\nu-d-\mu_2} \to \mathbb{K}[x]^3_{\nu-d} \to \mathbb{K}[x]_\nu \to \mathbb{K}[x]_\nu/(a, b, c)_\nu \to 0,$$

pour $\nu >> 0$. Comme la somme alternée des dimensions des $\mathbb{K}$-espaces vectoriels est nulle, et comme $\mathbb{K}[x]_\nu/(a,b,c)_\nu$ est 0 pour $\nu >> 0$, on a

$$\begin{aligned} 0 &= 3\,(d-\nu-1)+\nu-\mu_1-d+1+\nu-\mu_2-d+1+\nu+1 \\ &= d-\mu_1-\mu_2. \end{aligned}$$

□

Donc, pour $\mathcal{L}(\tilde{T}(x), x^n, x^{2n}-1)$, on a $\mu_1+\mu_2 = 2\,n$. On va démontrer, de plus que $\mu_1 = \mu_2 = n$:

Proposition 6.2.15. *Le $\mathbb{K}[x]$-module $\mathcal{L}(\tilde{T}(x), x^n, x^{2n}-1)$ admet une n-base.*

Démonstration. Considérons l'application suivante :

$$\begin{aligned} \mathbb{K}[x]^3_{n-1} &\to \mathbb{K}[x]_{3n-1} \\ (p(x),q(x),r(x)) &\mapsto \tilde{T}(x)p(x)+x^n q(x)+(x^{2n}-1)r(x). \end{aligned} \tag{6.6}$$

En faisant correspondre à chaque polynôme son vecteur des coefficients, cette application est équivalente à une application linéaire de $\mathbb{K}^{3n}$ dans $\mathbb{K}^{3n}$ dont sa matrice, S, est donnée par :

$$S := \left(\begin{array}{c|c|c} T_0 & \mathbf{0} & -\mathbb{I}_n \\ T_1 & \mathbb{I}_n & \mathbf{0} \\ T_2 & \mathbf{0} & \mathbb{I}_n \end{array}\right). \tag{6.7}$$

où T_0, T_1, T_2 sont les coefficients matricielles de $(\tilde{T}(x),\, x\,\tilde{T}(x),\, \ldots,\, x^n\tilde{T}(x))$, pour les listes des monômes $(1,\ldots,x^{n-1})$, $(x^n,\ldots,x^{2n-1})$, $(x^{2n},\ldots,x^{3n-1})$ respectivement. Remarquons qu'on a $T = T_0+T_2$.

En réduisant le bloc $(T_0|\mathbf{0}|-\mathbb{I}_n)$ par le bloc $(T_2|\mathbf{0}|\mathbb{I}_n)$, on peut donc le remplacer par $(T_0+T_2|\mathbf{0}|\mathbf{0})$, sans changer le rang de la matrice S. Comme $T = T_0+T_2$ est inversible, ça montre que S est de rang $3n$. C'est-à-dire $\ker(S) = 0$, ce qui est équivalent à dire qu'il n'y a pas des syzygies en degré $\leq n-1$.

Supposons que (p,q,r) et (p',q',r'), de degrés μ_1 et μ_2 respectivement, forment un système générateur de $\mathcal{L}(\tilde{T}(x), x^n, x^{2n}-1)$. Comme $2n = \mu_1+\mu_2$ et comme μ_1 et μ_2 sont $\geq n$ alors $\mu_1 = \mu_2 = n$. Donc il existe deux syzygies linéairement indépendants, (u_1,v_1,w_1) et (u_2,v_2,w_2) de degré n, qui engendrent $\mathcal{L}(\tilde{T}(x), x^n, x^{2n}-1)$. □

On peut trouver dans [129] un résultat similaire. La preuve est plus long que la notre, elle est basée sur des techniques d'interpolations et sur des calculs explicites.

On va décrire, dans la suite, comment construire explicitement, deux générateurs de $\mathcal{L}(\tilde{T}(x), x^n, x^{2n}-1)$ de degré n :

$\tilde{T}(x)$ est de degré $\leq 2\,n-1$ et l'application (6.6) est surjective, donc il existe $(u,v,w) \in \mathbb{K}[x]^3_{n-1}$ tel que

$$\tilde{T}(x)u(x)+x^n v(x)+(x^{2\,n}-1)\,w(x) = \tilde{T}(x)x^n, \tag{6.8}$$

on déduit donc, que $(u_1,v_1,w_1) = (x^n-u,-v,-w) \in \mathcal{L}(\tilde{T}(x), x^n, x^{2n}-1)$.

Pour les mêmes raisons, il existe $(u',v',w') \in \mathbb{K}[x]^3_{n-1}$ tel que

$$\tilde{T}(x)u'(x)+x^n v'(x)+(x^{2\,n}-1)\,w'(x) = 1 = x^n\,x^n-(x^{2\,n}-1) \tag{6.9}$$

on déduit donc, que $(u_2, v_2, w_2) = (-u', x^n - v', -w' - 1) \in \mathcal{L}(\tilde{T}(x), x^n, x^{2n} - 1)$.

De plus, les vecteurs (u_1, v_1, w_1) et (u_2, v_2, w_2) de $\mathcal{L}(\tilde{T}(x), x^n, x^{2n} - 1)$ sont linéairement indépendants, parce que, par construction, les coefficients vectoriels de x^n dans (u_1, v_1, w_1) et (u_2, v_2, w_2) sont $(1, 0, 0)$ et $(0, 1, 0)$ respectivement.

On peut maintenant, démontrer le théorème suivant qui donne la relation entre solution d'un système de Toeplitz et syzygies :

Théoreme 6.2.16. *Le vecteur u est solution de (6.1) si et seulement si il existe $v(x) \in \mathbb{K}[x]_{n-1}, w(x) \in \mathbb{K}[x]_{n-1}$ tels que*

$$(u(x), v(x), w(x)) \in \mathcal{L}(\tilde{T}(x), x^n, x^{2n} - 1; g(x))$$

Démonstration. D'après la proposition (6.2.5), le vecteur u est solution de (6.1) si est seulement si

$$\Pi_E(T(x)u(x)) = g(x).$$

Comme $u(x)$ est de degré $\leq n-1$, on déduit de (6.3) et (6.4) qu'il existe deux polynômes $A(x) \in \mathbb{K}[x]_{n-1}$ et $B(x) \in \mathbb{K}[x]_{n-1}$ tels que

$$T(x)u(x) - x^{-n+1}A(x) - x^n B(x) = g(x).$$

En évaluant cette équation aux racines $2n^{\text{ème}}$ de l'unité $\omega \in \mathfrak{U}_{2n}$, et comme $\omega^{-n} = \omega^n$ et $\tilde{T}(\omega) = T(\omega)$ pour $\omega \in \mathfrak{U}_n$, on aura

$$\tilde{T}(\omega)u(\omega) + \omega^n v(\omega) = g(\omega), \forall \omega \in \mathfrak{U}_{2n}(\omega), \tag{6.10}$$

avec $v(x) = -x\,A(x) - B(x)$ de degré $\leq n-1$.

L'équation (6.10) signifie que le polynôme $\tilde{T}(x)u(x) + x^n v(x) - g(x)$ s'annule sur tous les racines du polynôme $x^{2n} - 1$, ce qui signifie que $\tilde{T}(x)u(x) + x^n v(x) - g(x)$ est un multiple de $x^{2n} - 1$. On déduit donc qu'il existe un polynôme $w(x) \in \mathbb{K}[x]$ tel que

$$\tilde{T}(x)u(x) + x^n v(x) + (x^{2n} - 1)w(x) = g(x).$$

Remarquons de plus, que $w(x)$ est de degré $\leq n-1$, car $\tilde{T}(x)u(x)$ et $(x^{2n} - 1)\,w(x)$ sont de degré $\leq 3n - 1$.

Inversement, une solution $(u(x), v(x), w(x)) \in \mathcal{L}(\tilde{T}(x), x^n, x^{2n} - 1; g(x)) \cap \mathbb{K}[x]^3_{n-1}$ implique une solution $(u, v, w) \in \mathbb{K}^{3\,n}$ du système linéaire :

$$S \begin{pmatrix} u \\ v \\ w \end{pmatrix} = \begin{pmatrix} g \\ 0 \\ 0 \end{pmatrix},$$

où S est la matrice donnée par (6.7). On en déduit donc, que $T_2\,u + w = 0$ et que $T_0\,u - w = (T_0 + T_2)u = g$. Comme $T_0 + T_2 = T$, alors le vecteur u est la solution de (6.1). □

On a vu, quand on a étudié la formule de Gohberg-Semencul, que l'inverse d'une matrice de Teoplitz peut être reconstruit à partir de deux vecteurs seulement. Ces vecteurs

qui sont donnés dans (2.9) sont x et z, les solutions de $Tx = e_1$ et $Tz = ZTe_n$ respectivement. On a démontrer aussi que le vecteur y, tel que $Ty = e_n$, qui est utilisé, avec x, pour donner la formule de Gohberg-Semencul, est lié à x et z. Donc, x et z sont des générateurs de la matrice T^{-1}. On va démontrer dans la proposition suivante que, x et z ne sont que u et u' donnés par les équations (6.8) et (6.9). Donc, trouver une base de $\mathcal{L}(\tilde{T}(x), x^n, x^{2n} - 1)$ est équivalent à donner la décomposition de Gohberg-Semencul de T^{-1}. Et on va voir dans la suite, comment traduire la résolution du système linéaire, qui après décomposition de T^{-1} se fait par multiplication par des matrices de Toeplitz triangulaires, en une division euclidienne.

Proposition 6.2.17. *Soient $(u(x), v(x), w(x))$ et $(u'(x), v'(x), w'(x))$ dans $k_{n-1}[x]^3$, tels que*

$$\begin{cases} \tilde{T}(x)u(x) + x^n v(x) + (x^{2\,n} - 1)\, w(x) = \tilde{T}(x)x^n, \\ \tilde{T}(x)u'(x) + x^n v'(x) + (x^{2\,n} - 1)\, w'(x) = 1. \end{cases}$$

Alors $Tu' = e_1$ et $Tu = ZTe_n$.

Démonstration. Comme $u'(x)$, $v'(x)$, $w'(x)$ et 1 sont de degré $\leq n - 1$, alors, d'après le théorème (6.2.16), $\tilde{T}(x)u'(x) + x^n v'(x) + (x^{2\,n} - 1)\, w'(x) = 1$ équivalent à $Tu' = e_1$ $(e_1(x) = 1)$.

On a $\tilde{T}(x) = T_+(x) + x^{2n}T_-(x)$, donc,

$$\tilde{T}(x)u(x) + x^n v(x) + (x^{2n} - 1)w(x) = x^n T_+(x) + x^n((x^{2n} - 1)T_-(x) + T_-(x)),$$

par suite

$$\tilde{T}(x)u(x) + x^n(v(x) - T_+(x)) + (x^{2n} - 1)(w(x) - x^n T_-(x)) = x^n T_-(x).$$

comme $x^n T_-(x)$ est de degré $\leq n - 1$ et est le polynôme associé au vecteur ZTe_n, alors d'après le théorème (6.2.16), u est tel que $Tu = ZTe_n$. □

Du théorème (6.2.16) on peut déduire les deux corollaires suivants :

Corollaire 6.2.18. *Pour tout $g(x) \in \mathbb{K}_{n-1}[x]$, l'ensemble $\mathcal{L}(\tilde{T}(x), x^n, x^{2n} - 1; g) \cap \mathbb{K}^3_{n-1}[x]$ admet exactement un élément.*

Démonstration. Comme T est inversible, il existe un unique u tel que $Tu = g$. D'après le théorème (6.2.16), il existe $v(x)$, $w(x)$ de degrés $\leq n - 1$, tel que$(u(x), v(x), w(x)) \in \mathcal{L}(\tilde{T}(x), x^n, x^{2n} - 1; g) \cap \mathbb{K}^3_{n-1}[x]$.

L'unicité est évidente, parce que si $(u'(x), v'(x), w'(x)) \in \mathcal{L}(\tilde{T}(x), x^n, x^{2n} - 1; g) \cap \mathbb{K}^3_{n-1}[x]$, alors $(u(x), v(x), w(x)) - (u'(x), v'(x), w'(x)) \in \mathcal{L}(\tilde{T}(x), x^n, x^{2n} - 1) \cap \mathbb{K}^3_{n-1}[x]$ qui est égal $\{(0, 0, 0)\}$, voir la démonstration de la proposition (6.2.15). □

Corollaire 6.2.19. *Soit $\{(u_1, v_1, w_1), (u_2, v_2, w_2)\}$ une n-base de $\mathcal{L}(\tilde{T}(x), x^n, x^{2n} - 1)$. Soit (p, q, r) un élément de $\mathcal{L}(\tilde{T}(x), x^n, x^{2n} - 1; g(x))$. Il existe un unique $(u, v, w) \in \mathcal{L}(\tilde{T}(x), x^n, x^{2n} - 1; g) \cap \mathbb{K}^3_{n-1}[x]$ et deux polynômes p_1, p_2 uniques tels que*

$$\begin{pmatrix} p \\ q \\ r \end{pmatrix} = p_1 \begin{pmatrix} u_1 \\ v_1 \\ w_1 \end{pmatrix} + p_2 \begin{pmatrix} u_2 \\ v_2 \\ w_2 \end{pmatrix} + \begin{pmatrix} u \\ v \\ w \end{pmatrix}.$$

On va appeler cette décomposition une division de (p, q, r) par (u_1, v_1, w_1) et (u_2, v_2, w_2).

Démonstration. D'après le corollaire précédent, il existe un unique élément (u, v, w) dans $\mathcal{L}(\tilde{T}(x), x^n, x^{2n} - 1; g) \cap \mathbb{K}^3_{n-1}[x]$. Donc $(p, q, r) - (u, v, w)$ est dans $\mathcal{L}(\tilde{T}(x), x^n, x^{2n} - 1)$. Comme $\{(u_1, v_1, w_1), (u_2, v_2, w_2)\}$ est une n-base de $\mathcal{L}(\tilde{T}(x), x^n, x^{2n} - 1)$, alors il existe p_1, p_2 uniques tels que

$$\begin{pmatrix} p \\ q \\ r \end{pmatrix} - \begin{pmatrix} u \\ v \\ w \end{pmatrix} = p_1 \begin{pmatrix} u_1 \\ v_1 \\ w_1 \end{pmatrix} + p_2 \begin{pmatrix} u_2 \\ v_2 \\ w_2 \end{pmatrix}$$

□

On peut maintenant, démontrer simplement le théorème suivant :

Théoreme 6.2.20. *Soit $\{(u_1, v_1, w_1), (u_2, v_2, w_2)\}$ une n-base de $\mathcal{L}(\tilde{T}(x), x^n, x^{2n} - 1)$. Soit $g \in \mathbb{K}^n$. Le reste de la division, $(u(x), v(x), w(x))$, de $\begin{pmatrix} 0 \\ x^n g \\ g \end{pmatrix}$ par $\begin{pmatrix} u_1 & u_2 \\ v_1 & v_2 \\ w_1 & w_2 \end{pmatrix}$ est l'unique vecteur de $\mathcal{L}(\tilde{T}(x), x^n, x^{2n} - 1; g(x)) \cap \mathbb{K}^3_{n-1}[x]$ et par suite u est la solution de $Tu = g$.*

Démonstration. La démonstration est immédiate en utilisant les deux corollaires d'avant et en remarquant que $(0, x^n g(x), g(x)) \in \mathcal{L}(\tilde{T}(x), x^n, x^{2\,n} - 1; g)$ □

Une méthode pour faire la division demandée dans ce théorème est de choisir une n-base $\{(u_1, v_1, w_1), (u_2, v_2, w_2)\}$ de $\mathcal{L}(\tilde{T}(x), x^n, x^{2n} - 1)$, de façon que la matrice, 2×2 des coefficients de x^n dans

$$\begin{pmatrix} u_1(x) & u_2(x) \\ v_1(x) & v_2(x) \end{pmatrix},$$

soit inversible. Dans ce cas on peut réduire le polynôme $(0, x^n g(x))$ pour arriver à un degré $< n - 1$ et on peut écrire, d'une manière unique :

$$\begin{pmatrix} 0 \\ x^n g(x) \end{pmatrix} = p_1 \begin{pmatrix} u_1 \\ v_1 \end{pmatrix} + p_2 \begin{pmatrix} u_2 \\ v_2 \end{pmatrix} + \begin{pmatrix} u \\ v \end{pmatrix}.$$

A cause de l'unicité ici et l'unicité avant on obtient la proposition suivante :

Proposition 6.2.21. *La première coordonnée du reste vectoriel de la division de $\begin{pmatrix} 0 \\ x^n g \end{pmatrix}$ par $\begin{pmatrix} u & u_2 \\ v_1 & v_2 \end{pmatrix}$ est le polynôme $u(x)$ tel que son vecteur associé u est la solution de $Tu = g$.*

Maintenant, après cette étude théorique, on va décrire comment calculer une n- base de $\mathcal{L}(\tilde{T}(x), x^n, x^{2n} - 1)$, puis comment on fait la division demandée dans la propositon précédente.

6.2.2 Construction des générateurs

Définition 6.2.22. *On note par $\sigma_1, \sigma_2, \sigma_3$ la base canonique de $\mathbb{K}^3[x]$ considéré comme $\mathbb{K}[x]$-module. Et on note par ρ_1, ρ_2 la n-base de $\mathcal{L}(\tilde{T}(x), x^n, x^{2n} - 1)$ donnée, dans* (6.8) *et* (6.9)*, par*

$$\begin{aligned} \rho_1 &= x^n \sigma_1 - (u, v, w) = (u_1, v_1, w_1) \\ \rho_2 &= x^n \sigma_2 - (u', v', w') = (u_2, v_2, w_2) \end{aligned} \tag{6.11}$$

On va décrire, dans la suite, comment calculer ρ_1 et ρ_2. On donnera deux méthodes pour les calculer : La deuxième méthode et celle donnée dans [129]. La première, est une méthode qui utilise l'algorithme euclidien de calcul de PGCD de deux polynômes, qu'on appelle algorithme d'Euclide étendu.

Commençons par un rappel sur quelques propriétés théoriques et pratiques de l'algorithme d'Euclide étendu, pour plus d'informations voir [134] :

Soient $p(x)$ et $p'(x)$ deux polynômes de degré m et m' respectivement. Soient

$$\begin{array}{ll} r_0 = p, & r_1 = p', \\ s_0 = 1, & s_1 = 0, \\ t_0 = 0, & t_1 = 1. \end{array}$$

et définissons les relations de récurrences suivantes :

$$\begin{aligned} r_{i+1} &= r_{i-1} - q_i r_i, \\ s_{i+1} &= s_{i-1} - q_i s_i, \\ t_{i+1} &= t_{i-1} - q_i t_i, \end{aligned}$$

où q_i est la quotient de la division de r_{i-1} par r_i, c'est-à-dire $r_{i-1} = q_i r_i + r_{i+1}$.

Proposition 6.2.23. *Soit $l \in \mathbb{N}$ tel que $r_l = 0$. Alors $r_{l-1} = \gcd(p(x), p'(x))$.*

Et plus généralement, si on s'arrête avant d'arriver à $r_l = 0$, on a la proposition suivante :

Proposition 6.2.24. *Pour tout $i = 1, \ldots, l$, on a*

$$s_i p + t_i p' = r_i \quad \text{et} \quad (s_i, t_i) = 1, \tag{6.12}$$

de plus on a

$$\begin{cases} \deg r_{i+1} < \deg r_i, \quad i = 1, \ldots, l-1 \\ \deg s_{i+1} > \deg s_i \quad \text{et} \quad \deg t_{i+1} > \deg t_i, \\ \deg s_{i+1} = \deg(q_i.s_i) = \deg v - \deg r_i, \\ \deg t_{i+1} = \deg(q_i.t_i) = \deg u - \deg r_i. \end{cases}$$

On peut, maintenant, présenter notre algorithme. Il est donné dans la preuve du théorème suivant :

Théoreme 6.2.25. *En utilisant l'algorithme d'Euclide étendu sur $p(x) = x^{n-1}T(x)$ et $p'(x) = x^{2n-1}$ et en s'arrêtant en degré $n-1$ et $n-2$ on obtient ρ_1 et ρ_2 respectivement.*

Démonstration. On a vu que si $Tu = g$ alors il existe deux polynômes $A(x)$ et $B(x)$ dans $\mathbb{K}_{n-1}[x]$ tels que

$$\bar{T}(x)u(x) + x^{2n-1}B(x) = x^{n-1}g(x) + A(x),$$

où $\bar{T}(x) = x^{n-1}T(x)$ est un polynôme de degré $\leq 2n-2$. On a vu dans (6.8) et (6.9) que pour $g(x) = 1$ ($g = e_1$) et $g(x) = x^n T(x)$ ($g = (0, t_{-n+1}, \ldots, t_{-1})^T$) on obtient une n-base de $\mathcal{L}(\tilde{T}(x), x^n, x^{2n} - 1)$.

Donc $Tu_1 = e_1$ si et seulement si il existe $A_1(x)$, $B_1(x)$ tels que

$$\bar{T}(x)u_1(x) + x^{2n-1}B_1(x) = x^{n-1} + A_1(x) \tag{6.13}$$

et $Tu_2 = (0, t_{-n+1}, \ldots, t_{-1})^T$ si et seulement si il existe $A_2(x)$, $B_2(x)$ tels que

$$\bar{T}(x)(u_2(x) + x^n) + x^{2n-1}B_2(x) = A_2(x), \tag{6.14}$$

avec $\deg A_1(x) \leq n-2$ et $\deg A_2(x) \leq n-2$.

L'équation (6.13) est de la forme (6.12), avec $p(x) = \bar{T}(x)$ et $p'(x) = x^{2n-1}$. En démontrant l'unicité de cette écriture et en remarquant qu'elle peut être obtenue par application de l'algorithme d'Euclide à l'ordre $n-1$. On fait les mêmes remarques pour (6.14). Donc en appliquant l'algorithme d'Euclide étendu sur $p(x) = x^{n-1}T$ et $p'(x) = x^{2n-1}$ jusqu'on a $\deg r_l(x) = n-1$ et $\deg r_{l+1}(x) = n-2$ on obtient

$$u_1(x) = \frac{1}{c_1}s_l(x), \quad B_1(x) = \frac{1}{c_1}t_l(x), \quad x^{n-1} + A_1(x) = \frac{1}{c_1}r_l(x)$$

et

$$x^n + u_2(x) = \frac{1}{c_2}s_{l+1}(x), \quad B_2(x) = \frac{1}{c_2}t_{l+1}(x), \quad A_2(x) = \frac{1}{c_2}r_{l+1}(x)$$

avec c_1 et c_2 sont les coefficients de plus grand degré de $r_l(x)$ et $s_{l+1}(x)$ respectivement. En effet : l'équation (6.13) est équivalent à l'écriture matricielle suivante

$$\begin{array}{c} \\ n-1 \\ \\ n \\ \\ n-1 \\ \\ \end{array}
\overbrace{\phantom{\begin{pmatrix} a & b & c \end{pmatrix}}}^{n}
\overbrace{\phantom{\begin{pmatrix} a & b \end{pmatrix}}}^{n-1}
\left(\begin{array}{ccc|c}
t_{-n+1} & & & \\
\vdots & \ddots & & \\
\hline
t_0 & \ldots & t_{-n+1} & \\
\vdots & \ddots & \vdots & \\
t_{n-1} & \ldots & t_0 & \\
\hline
 & \ddots & \vdots & 1 \\
 & & & \ddots \\
 & & t_{n-1} & & 1
\end{array}\right)
\begin{pmatrix} u_1 \\ \\ B_1 \end{pmatrix}
=
\left(\begin{array}{c} A_1 \\ \hline 1 \\ 0 \\ \vdots \\ 0 \end{array}\right).$$

Comme T est inversible, le bloc de taille $(2n-1)\times(2n-1)$ qui est en bas de la matrice sera inversible. Donc, u_1 et B_1 sont uniques et par suite A_1 sera aussi unique.

Comme $\deg r_l = n-1$, alors par la proposition (6.2.24), $\deg s_{l+1} = (2n-1)-(n-1) = n$ et $\deg t_{l+1} = (2n-2)-(n-1) = n-1$. Par la même proposition on a aussi : $\deg s_l \leq n-1$ et $\deg t_l \leq n-2$.

Donc u_1 et s_l ont le même degré, et B_1 et t_l ont le même degré. Par suite, à cause de l'unicité, et comme u_1 est unitaire, $\frac{1}{c_1}s_l = u_1$ et $\frac{1}{c_1}t_l = B_1$. De même on a que $x^n + u_2(x) = \frac{1}{c_2}s_{l+1}(x)$ et $B_2(x) = \frac{1}{c_2}t_{l+1}(x)$.

Finalement, $Tu = e_1$ si et seulement si il existe $v(x)$, $w(x)$ tels que

$$\tilde{T}(x)u(x) + x^n v(x) + (x^{2n}-1)w(x) = 1. \tag{6.15}$$

Comme $\tilde{T}(x) = T_+ + x^{2n}T_- = T + (x^{2n}-1)T_-$ alors

$$T(x)u(x) + x^n v(x) + (x^{2n}-1)(w(x) + T_-(x)u(x)) = 1. \tag{6.16}$$

D'une autre côté, on a $T(x)u(x) - x^{-n+1}A_1(x) + x^n B_1(x) = 1$ et $x^{-n+1}A_1(x) = x^n(xA_1) - x^{-n}(x^{2n}-1)xA_1$. On a donc

$$T(x)u(x) + x^n(B(x) - xA(x)) + (x^{2n}-1)x^{-n+1}A(x) = 1. \tag{6.17}$$

En comparant (6.16) et (6.17), et comme $1 = x^n x^n - (x^{2n}-1)$ on achève la démonstration et on a de plus $w(x) = x^{-n+1}A(x) - T_-(x)u(x) + 1$ qui est la partie des termes de degré positif de $-T_-(x)u(x) + 1$. □

L'algorithme d'Euclide étendue est un algorithme d'ordre $\mathcal{O}(n^2)$, mais un algorithme rapide qui n'utilise pas plus que $\mathcal{O}(nlog^2 n)$, est donné dans [134] chapitre 11.

La deuxième méthode pour calculer (u_1, v_1, w_1) et (u_2, v_2, w_2) est donné dans [129]. On va le décrire rapidement ici : Cet algorithme calcule les coefficients de σ_1, σ_2 d'une n-base de $\mathcal{L}(\tilde{T}, x^n, x^{2n}-1)$. Les coefficients de σ_3 correspondent aux éléments dans l'idéal $(x^{2n}-1)$, on peut, donc, en réduisant $(\tilde{T}(x)\, x^n).B(x)$ par $x^{2n}-1$, avec

$$B(x) = \begin{pmatrix} x^n - u_0 & -v_0 \\ -u_1 & x^n - v_1 \end{pmatrix} = \begin{pmatrix} u(x) & v(x) \\ u'(x) & v'(x) \end{pmatrix},$$

où (u, v, w) et (u', v', w') sont les générateurs donnés par (6.8) et (6.9).

On commence par décrire l'algorithme :

En évaluant (6.11) aux $\omega_j \in \mathfrak{U}_{2n}$, on peut déduire que $(u(x)\, v(x))^T$ et $(u'(x)\, v'(x))^T$ sont la solution du problème d'interpolation suivant :

$$\begin{cases} \tilde{T}(\omega_j)u(\omega_j) + \omega_j^n v(\omega_j) = 0 \\ \tilde{T}(\omega_j)u'(\omega_j) + \omega_j^n v'(\omega_j) = 0 \end{cases} \quad \text{avec}$$

$$\begin{cases} u_n = 1,\ v_n = 0 \\ u'_n = 0,\ v'_n = 1 \end{cases}$$

On va utiliser la notation suivante :

Définition 6.2.26. *Le τ-degré d'un vecteur de polynômes $d(x) = (d_1(x)\, d_2(x))^T$ est défini par*

$$\tau - \deg d(x) := \max\{\deg d_1(x),\ \deg d_2(x) - \tau\}$$

Définition 6.2.27. *Un ensemble de vecteurs de polynômes dans $\mathbb{K}[x]^2$ est appelé τ-réduit, si les vecteurs des coefficients de plus grand τ-degré sont linéairement indépendants.*

Remarquons que $B(x)$ est une $n-$base réduite du module de tous les vecteurs $r(x) \in \mathbb{K}[x]^2$ qui vérifient les conditions d'interpolation suivantes :

$$f_j^T r(\omega_j) = 0, \quad j = 0, \ldots, 2n-1$$

avec $f_j = \begin{pmatrix} \tilde{T}(\omega_j) \\ \omega_j^n \end{pmatrix}$. $B(x)$ sera appelé une τ-base réduite associée aux donnés d'interpolation $(\omega_j, f_j),\ j = 0, \ldots, 2n-1$.

Théoreme 6.2.28. *Soit $\tau = n$. Soit J un entier positif. Soient $\sigma_1, \ldots, \sigma_J \in \mathbb{K}$ et $\phi_1, \ldots, \phi_J \in \mathbb{K}^2$ qui sont $\neq (0\,0)^T$. Soient $1 \leq j \leq J$ et $\tau_J \in \mathbb{Z}$. Supposons que $B_j(x) \in \mathbb{K}[x]^{2\times 2}$ est une base τ_J-base réduite associée aux donnés d'interpolation $\{(\sigma_i, \phi_i); i = 1, \ldots, j\}$, où les colonnes de $B_j(x)$ ont comme τ_J-degré δ_1 et δ_2 respectivement.*

Soit $\tau_{j\to J} = \delta_1 - \delta_2$. Soit $B_{j\to J}(x)$ une $\tau_{j\to J}$-base réduite associée aux donnés d'interpolation $\{(\sigma_i, B_j^T(\sigma_j)\phi_i); i = j+1, \ldots, J\}$.

Alors $B_J(x) := B_j(x)B_{j\to J}(x)$ est une τ_J-base réduite associée aux donnés d'interpolation $\{(\sigma_i, \phi_i); i = 1, \ldots, J\}$.

Démonstration. Voir [129] pour la démonstration de ce théorème. □

En appliquant ce théorème sur $\omega_j \in \mathfrak{U}_{2n}$ comme points d'interpolation, on obtient un algorithme rapide en $\mathcal{O}(n \log^2 n)$ qui calcule $B(x)$, voir [129].

6.2.3 Division Euclidien

On va décrire dans cette section, comment diviser rapidement, $\begin{pmatrix} 0 \\ x^n\, g \\ g \end{pmatrix}$ par $\begin{pmatrix} u_1 & u_2 \\ v_1 & v_2 \\ w_1 & w_2 \end{pmatrix}$ pour obtenir l'unique élément de $\mathcal{L}(\tilde{T}(x), x^n, x^{2n} - 1; g) \cap \mathbb{K}^3_{n-1}[x]$. Et on a vu, dans la proposition (6.2.21), qu'il suffit de diviser $\begin{pmatrix} 0 \\ x^n g \end{pmatrix}$ par $\begin{pmatrix} u_1 & u_2 \\ v_1 & v_2 \end{pmatrix}$. On pose donc, le problème suivant :

Problème 6.2.29. *Soit la matrice des polynômes $\begin{pmatrix} e(x) & e'(x) \\ f(x) & f'(x) \end{pmatrix}$ de degré n telle que $\begin{pmatrix} e_n & e'_n \\ f_n & f'_n \end{pmatrix}$ est inversible. Soit le vecteur des polynômes $\begin{pmatrix} p(x) \\ q(x) \end{pmatrix}$ de degré $m \geq n$. Trouver le reste de la division de $\begin{pmatrix} p(x) \\ q(x) \end{pmatrix}$ par $\begin{pmatrix} e(x) & e'(x) \\ f(x) & f'(x) \end{pmatrix}$.*

On va décrire ici un algorithme d'Euclide généralisé qui résout le problème (6.2.29).

On va noter $E(x) = \begin{pmatrix} p(x) \\ q(x) \end{pmatrix}$ et $B(x) = \begin{pmatrix} e(x) & e'(x) \\ f(x) & f'(x) \end{pmatrix}$. Écrivons $E(x) = B(x)Q(x) + R(x)$ avec $\deg(R(x)) < n$, et $\deg(Q(x)) \leq m - n$ et posons $z = \frac{1}{x}$. On a

$$\begin{aligned} & E(x) && = B(x)Q(x) + R(x) \\ \Leftrightarrow \quad & E(\frac{1}{z}) && = B(\frac{1}{z})Q(\frac{1}{z}) + R(\frac{1}{z}) \\ \Leftrightarrow \quad & z^m E(\frac{1}{z}) && = z^n B(\frac{1}{z}) z^{m-n} Q(\frac{1}{z}) + z^{m-n+1} z^{n-1} R(\frac{1}{z}) \\ \Leftrightarrow \quad & \hat{E}(z) && = \hat{B}(z)\hat{Q}(z) + z^{m-n+1}\hat{R}(z) \end{aligned} \tag{6.18}$$

avec $\hat{E}(z), \hat{B}(z), \hat{Q}(z), \hat{R}(z)$ sont les polynômes obtenus en inversant l'ordre des coefficients des polynômes $E(z), B(z), Q(z), R(z)$ respectivement. Comme le terme constant de $\hat{Q}(z)$

est non nul, alors, $\hat{Q}(z)$ vu comme une série formelle, est inversible. On a donc,

$$\begin{aligned} (6.18) \quad \Rightarrow \quad & \frac{\hat{E}(z)}{\hat{B}(z)} = \hat{Q}(z) + z^{m+n-1}\frac{\hat{R}(z)}{\hat{B}(z)} \\ \Rightarrow \quad & \hat{Q}(z) = \frac{\hat{E}(z)}{\hat{B}(z)} \mod z^{m-n+1} \end{aligned}$$

Par suite $\hat{Q}(z)$ est obtenu en calculant les premiers $m-n+1$ coefficients de $\dfrac{\hat{E}(z)}{\hat{B}(z)}$, qu'on va le calculer en calculant $W(x) = \dfrac{1}{\hat{B}(z)}$, puis en multipliant $W(x)$ par $\hat{E}(z)$.

On utilisera les itérations de Newton pour calculer $W(x)$. Soit $f(W) = \hat{B} - W^{-1}$. $f'(W_l).(W_{l+1} - W_l) = -W_l^{-1}(W_{l+1} - W_l)W_l^{-1} = f(W_l) = \hat{B} - W_l^{-1}$, donc

$$W_{l+1} = 2W_l - W_l\hat{B}W_l.$$

et $W_0 = \hat{B}_0^{-1}$.

$$\begin{aligned} W - W_{l+1} &= W - 2W_l + W_l\hat{B}W_l \\ &= W(I_2 - \hat{B}W_l)^2 \\ &= (W - W_l)\hat{B}(W - W_l) \end{aligned}$$

Donc $W_l(x) = W(x) \mod x^{2l}$ pour $l = 0, \ldots, \lceil \log(m-n+1) \rceil$.

Proposition 6.2.30. *On a besoin de* $\mathcal{O}(n\log(n)\log(m-n) + m\log m)$ *ops pour résoudre le problème* (6.2.29).

Démonstration. On a besoin de faire $\lceil \log(m-n+1) \rceil$ itérations pour obtenir les $m-n+1$ premiers coefficients de $\dfrac{1}{\hat{B}} = W(x)$. Pour chaque itération, on doit multiplier deux polynômes en degré n, ce qui coûte $\mathcal{O}(n\log n)$ ops. Finalement $\mathcal{O}(m\log m)$ ops sont nécessaires pour multiplier $\hat{E}$ par $\dfrac{1}{\hat{B}}$. □

6.3 Cas de deux variables

Dans cette section, on va essayer de généraliser ce qu'on a fait, pour le cas d'une matrice de Toeplitz scalaire et polynômes en une seule variable, au cas d'une matrice de Toeplitz par blocs de Toeplitz et polynômes en deux variables.

Soit $m \in \mathbb{N}$ et $n \in \mathbb{N}$. Dans cette section, on note par $E = \{(i,j) \in \mathbb{N}^2;\ 0 \leq i \leq m-1,\ 0 \leq j \leq n-1\}$ et $R = \mathbb{K}[x,y]$. On note par $\mathbb{K}[x,y]_n^m$ l'ensemble des polynômes en deux variables de degré $\leq m$ en x et $\leq n$ en y. Pour $p_1, \ldots, p_k$ dans R, on dénote par $(p_1, \ldots, p_k)$ l'idéal engendré par $p_1, \ldots, p_k$.

Notation 6.3.1. *Pour une matrice M formée de m blocs de taille $n \times n$, on utilisera la notation suivante :*

$$M = \left(M_{(i_1,i_2),(j_1,j_2)}\right)_{\substack{0\le i_1,j_1\le m-1\\ 0\le i_2,j_2\le n-1}} = (M_{\alpha\beta})_{\alpha,\beta\in E}. \tag{6.19}$$

(i_1, j_1) donnent les positions des blocs, (i_2, j_2) donnent les positions dans les blocs.

Problème 6.3.2. *Soit*

$$T = (t_{\alpha-\beta})_{\alpha\in E,\beta\in E} \in \mathbb{K}^{mn\times mn} \tag{6.20}$$

($T = (T_{\alpha\beta})_{\alpha,\beta\in E}$ avec $T_{\alpha\beta} = t_{\alpha-\beta}$) une matrice de Toeplitz par blocs de Toeplitz de taille $mn \times mn$, elle est formée de m blocs de taille $n \times n$. Soit g un vecteur de longueur mn et qu'on l'indexe de la manière suivante :

$$g = (g_\alpha)_{\alpha\in \mathtt{E}} \in \mathbb{K}^{mn}. \tag{6.21}$$

Le problème est de trouver

$$u = (u_\alpha)_{\alpha\in \mathtt{E}}, \tag{6.22}$$

tel que

$$Tu = g \tag{6.23}$$

On commence par définir les polynômes en deux variables suivants :

Définition 6.3.3. *Pour T, g et u définis dans* (6.20), (6.21) *et* (6.22) *respectivement, on définit les polynômes associés suivants :*

- $T(x,y) = \sum_{\substack{(i,j)\in \mathtt{E}-\mathtt{E}\\ 2n-1,2m-1}} t_{i,j}x^iy^j,$
- $\tilde{T}(x,y) = \sum_{i,j=0} \tilde{t}_{i,j}x^iy^j$ *avec*

$$\tilde{t}_{i,j} := \begin{cases} t_{i,j} & si\ i < m, j < n \\ t_{i-2m,j} & si\ i \ge m, j < n \\ t_{i,j-2n} & si\ i < m, j \ge n \\ t_{i-2m,j-2n} & si\ i \ge m, i \ge n \end{cases},$$

- $u(x,y) = \sum_{(i,j)\in \mathtt{E}} u_{i,j}\, x^iy^j,$
- $g(x,y) = \sum_{(i,j)\in \mathtt{E}} g_{i,j}x^iy^j.$

Remarquons qu' on peut décomposer $T(x,y)$ de la façon suivante :

$$T(x,y) = T_{++}(x,y) + T_{-+}(x,y) + T_{+-}(x,y) + T_{--}(x,y),$$

avec $T_{++}(x,y)$ contient les termes de degré positif en x et y, $T_{-+}(x,y)$ contient les termes de degré négatif en x et positif en y, $T_{+-}(x,y)$ contient les termes de degré positif en x et négatif en y et $T_{--}(x,y)$ contient les termes de degré négatif en x et y. On a alors la remarque suivante :

Remarque 6.3.4. *On a*

$$\tilde{T}(x,y) = T_{++}(x,y) + x^{2m}T_{-+}(x,y) + y^{2n}T_{+-}(x,y) + x^{2m}y^{2n}T_{--}(x,y).$$

6.3.1 Matrices de Toeplitz par blocs de Toeplitz et syzygies

Définition 6.3.5. *Soit* $\mathbf{a} = (a_1, \ldots, a_k) \in \mathbb{K}[x,y]^k$ *un vecteur, non nul, des polynômes en deux variables. On dénote par* $\mathcal{L}(\mathbf{a})$ *l'ensemble des vecteurs* $(h_1, \ldots, h_k) \in \mathbb{K}[x,y]^k$ *tel que*

$$\sum_{i=1}^{n} a_i\, h_i = 0. \tag{6.24}$$

$\mathcal{L}(\mathbf{a})$ *est l'ensemble des syzygies de* $\mathbf{a}$.

L'ensemble $\mathcal{L}(\mathbf{a})$ est un $\mathbb{K}[x,y]$-module de $\mathbb{K}[x,y]^n$.

Théoreme 6.3.6. *Le vecteur* u *est solution de* (6.23) *si est seulement si il existe*

$$h_2, \ldots, h_9 \in \mathbb{K}[x,y]_{n-1}^{m-1}$$

tels que $(u(x,y), h_2(x,y), \ldots, h_9(x,y))$ *soit dans*

$$\mathcal{L}(\tilde{T}(x,y), x^m, x^{2m}-1, y^n, x^m y^n, (x^{2m}-1)y^n, y^{2n}-1, x^m(y^{2n}-1), (x^{2m}-1)(y^{2n}-1); g).$$

Démonstration. Soit $L = \{x^{\alpha_1} y^{\alpha_2}, 0 \le \alpha_1 \le m-1, 0 \le \alpha_2 \le n-1\}$ l'ensemble des monômes de degré $\le m-1$ en x et $\le n-1$ en y. Soit Π_E la projection de R sur l'espace vectoriel engendré par L.

Par [88] proposition (3.5.3), on a

$$Tu = g \Leftrightarrow \Pi_E(T(x,y)\, u(x,y)) = g(x,y). \tag{6.25}$$

Or $T(x,y)u(x,y)$ est un polynôme de degré entre $-m+1$ et $2m-2$ en x et entre $-n+1$ et $2n-2$ en y. On peut donc le décomposer de la façon suivante :

$$\begin{aligned} T(x,y)u(x,y) = \Pi_E(T(x,y)u(x,y)) + x^m y^n A_1(x,y) + x^m y^{-n} A_2(x,y) + x^{-m} y^n A_3(x,y) + \\ x^{-m} y^{-n} A_4(x,y) + x^m A_5(x,y) + x^{-m} A_6(x,y) + y^n A_7(x,y) + y^{-n} A_8(x,y), \end{aligned}$$

où, les $A_i(x,y)$ $i = 1, \ldots, 8$ sont des polynômes de degré au plus $m-1$ en x et $n-1$ en y. Par suite (6.25) donne l'équation suivante :

$$\begin{aligned} T(x,y)u(x,y) = g(x,y) + x^m y^n A_1(x,y) + x^m y^{-n} A_2(x,y) + x^{-m} y^n A_3(x,y) + \\ x^{-m} y^{-n} A_4(x,y) + x^m A_5(x,y) + x^{-m} A_6(x,y) + y^n A_7(x,y) + y^{-n} A_8(x,y), \end{aligned} \tag{6.26}$$

Soit $\omega \in \mathfrak{U}_{2m}$ et $\upsilon \in \mathfrak{U}_{2n}$. On a $\omega^m = \omega^{-m}$ et $\upsilon^n = \upsilon^{-n}$. De plus, d'après la remarque (6.3.4),

$$\tilde{T}(\omega,\upsilon) = T_{++}(\omega,\upsilon) + T_{-+}(\omega,\upsilon) + T_{+-}(\omega,\upsilon) + T_{--}(\omega,\upsilon) = T(\omega,\upsilon).$$

Donc, en tenant compte de ces remarques et en évaluant l'équation (6.26) en (ω,υ) on trouve que

$$\tilde{T}(\omega,\upsilon)u(\omega,\upsilon) + x^m h_2(\omega,\upsilon) + y^n h_4(\omega,\upsilon) + x^m y^n h_5(\omega,\upsilon) - g(\omega,\upsilon) = 0,$$

avec $h_2 = -(A_5+A_6)$, $h_4 = -(A_7+A_8)$, et $h_5 = -(A_1(x,y)+A_2(x,y)+A_3(x,y)+A_4(x,y))$. Ça signifie que

$$p(x,y)=\tilde{T}(x,y)u(x,y)+x^m h_2(x,y)+y^n h_4(x,y)+x^m y^n h_5(x,y)-g(x,y) \in (x^{2m}-1, y^{2n}-1).$$

En réduisant par les polynômes $x^{2m}-1$ et $y^{2n}-1$, et comme $p(x,y)$ est de degré $\leq 3m-1$ en x et $\leq 3n-1$ en y, on peut déduire l'existence de $h_3(x,y), h_6(x,y), \ldots, h_8(x,y) \in \mathbb{K}[x,y]_{n-1}^{m-1}$ tels que

$$\begin{aligned}&\tilde{T}(x,y)u(x,y) + x^m\, h_2(x,y) + (x^{2m}-1)h_3(x,y) + y^n h_4(x,y) + x^m y^n h_5(x,y)\\ &+(x^{2m}-1)y^n h_6(x,y) + (y^{2n}-1)h_7(x,y) + x^m(y^{2m}-1)h_7(x,y)\\ &+(x^{2n}-1)(y^{2n}-1)h_8(x,y) = g(x,y) \qquad (6.27)\end{aligned}$$

Inversement, une solution de (6.27) peut être transformée en une solution de (6.26). L'idée de la transformation est la même idée utilisée pour le cas scalaire et qui est donnée dans la démonstration du théorème (6.2.16). On utilise la matrice S donnée dans la preuve de la proposition, (6.3.8), suivante. □

Dans la suite, on utilisera la notation suivante :

Définition 6.3.7. *On dénote par* $\mathbf{T}$ *le vecteur* $\mathbf{T} = (\tilde{T}(x,y), x^m, x^{2m}-1, y^n, x^m y^n, (x^{2m}-1)y^n, y^{2n}-1, x^m(y^{2n}-1), (x^{2m}-1)(y^{2n}-1))$.

Comme dans le scalaire, on peut démontrer ici, qu'il n'y a pas de syzygies en degré $\leq m-1$ en x et $\leq n-1$ en y :

Proposition 6.3.8. *Il n'y a pas d'élément non nul dans l'ensemble* $\mathcal{L}(\mathbf{T}) \cap \mathbb{K}[x,y]_{n-1}^{m-1}{}^{9}$.

Démonstration. On considère l'application suivante :

$$\begin{aligned}\mathbb{K}[x,y]_{n-1}^{m-1}{}^{9} &\to \mathbb{K}[x,y]_{3n-1}^{3m-1} &(6.28)\\ p(x,y) = (p_1(x,y), \ldots, p_9(x,y)) &\mapsto \mathbf{T}.p &(6.29)\\ & &(6.30)\end{aligned}$$

Sa matrice, de taille $9mn \times 9mn$, est de la forme suivante :

$$S = \left(\begin{array}{c|ccc|ccc|ccc} & E_{21} & -E_{11}+E_{31} & & & & & -E_{11} & -E_{21} & E_{11}-E_{31}\\ T_0 & \vdots & \vdots & & & & & \vdots & \vdots & \vdots\\ & E_{2m} & -E_{1m}+E_{3m} & & & & & -E_{1m} & -E_{2m} & E_{1m}-E_{3m}\\ \hline & & & E_{11} & E_{21} & -E_{11}+E_{31} & & & &\\ T_1 & & & \vdots & \vdots & \vdots & & & &\\ & & & E_{1m} & E_{2m} & -E_{1m}+E_{3m} & & & &\\ \hline & & & & & & E_{11} & E_{21} & -E_{11}+E_{31}\\ T_2 & & & & & & \vdots & \vdots & \vdots\\ & & & & & & E_{1m} & E_{2m} & -E_{1m}+E_{3m}\end{array}\right) \qquad (6.31)$$

avec E_{ij} est la matrice de taille $3n \times mn$ $e_{ij} \otimes I_n$ où e_{ij} la matrice de taille $3 \times m$ avec des coefficients nuls sauf dans la position (i,j) le coefficient vaut 1. Et $\begin{pmatrix} T_0 \\ T_1 \\ T_2 \end{pmatrix}$ est la matrice, de taille $9mn \times n$, suivante :

$$\left(\begin{array}{cccc} t_0 & 0 & \dots & 0 \\ t_1 & t_0 & \dots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ t_{m-1} & \dots & t_1 & t_0 \\ \hline 0 & t_{m-1} & \dots & t_1 \\ t_{-m+1} & 0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & t_{-m+1} \\ t_{-1} & \dots & t_{-m+1} & 0 \\ \hline 0 & t_{-1} & \dots & t_{-m+1} \\ \vdots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & t_{-1} \\ 0 & \dots & \dots & 0 \end{array}\right) \text{ et } t_i = \left(\begin{array}{cccc} t_{i,0} & 0 & \dots & 0 \\ t_{i,1} & t_{i,0} & \dots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ t_{i,n-1} & \dots & t_{i,1} & t_{i,0} \\ \hline 0 & t_{i,n-1} & \dots & t_{i,1} \\ t_{i,-n+1} & 0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & t_{i,-n+1} \\ t_{i,-1} & \dots & t_{i,-n+1} & 0 \\ \hline 0 & t_{i,-1} & \dots & t_{i,-n+1} \\ \vdots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & t_{i,-1} \\ 0 & \dots & \dots & 0 \end{array}\right)$$

Pour des raisons semblables à celles données dans la démonstration de la proposition (6.2.15), en faisant des réductions dans les blocs puis par blocs, la matrice S est inversible. C'est à dire $\ker S = \{0\}$, ce qui prouve la proposition. □

Corollaire 6.3.9. *L'ensemble $\mathcal{L}(\mathbf{T}; g) \cap \mathbb{K}[x,y]_{\substack{m-1\\n-1}}^{9}$ contient exactement un seul élément.*

Démonstration. Comme T est inversible, il existe un unique u tel que $Tu = g$. D'après le théorème (6.3.6), il existe $h_2, \dots, h_9 \in \mathbb{K}[x,y]_{\substack{m-1\\n-1}}$ tels que $U = (u, h_2, \dots, h_9) \in \mathcal{L}(\mathbf{T}; g)$.

L'unicité est évidente, parce que si $U' \in \mathcal{L}(\mathbf{T}; g) \cap \mathbb{K}[x,y]_{\substack{m-1\\n-1}}^{9}$ alors $U - U' \in \mathcal{L}(\mathbf{T}) \cap \mathbb{K}[x,y]_{\substack{m-1\\n-1}}^{9} = \{0\}$ d'après la proposition précédente. □

6.3.2 Générateurs et réduction

On commence par démontrer le théorème suivant :

Théoreme 6.3.10. *Pour tout vecteur de polynômes $\mathbf{a} = (a_i)_{i=1,\dots,n} \in \mathbb{K}[x,y]^n$, le $\mathbb{K}[x,y]$-module $\mathcal{L}(a_1, \dots, a_n)$ est libre de rang $n-1$.*

Démonstration. Considérons tout d'abord, le cas où les a_i sont des monômes :

$a_i = x^{\alpha_i} y^{\beta_i}$, qu'on va ordonner dans l'ordre lexicographique tel que $x < y$. Et on va supposer que $a_1 \geq \dots \geq a_n$. Dans ce cas, le module de syzygies de $\mathbf{a}$ est engendré par les S-polynômes :

$$S(a_i, a_j) = \operatorname{lcm}(a_i, a_j)\left(\frac{\sigma_i}{a_i} - \frac{\sigma_j}{a_j}\right),$$

où $(\sigma_i)_{i=1,\dots,n}$ est la base canonique de $\mathbb{K}[x,y]^n$, voir [37].

On peut vérifier simplement que :

$$S(a_i,a_k)=\frac{\operatorname{lcm}(a_i,a_k)}{\operatorname{lcm}(a_i,a_j)}S(a_i,a_j)-\frac{\operatorname{lcm}(a_i,a_k)}{\operatorname{lcm}(a_j,a_k)}S(a_j,a_k) \text{ if } i\geq j\geq k$$

et que $\operatorname{lcm}(a_i,a_j)$ divise $\operatorname{lcm}(a_i,a_k)$. Donc $\mathcal{L}(\mathbf{a})$ est engendré par les $S(a_i,a_j)$ qui sont minimales pour la division, c'est-à-dire par $S(a_i,a_{i+1})$ pour $i=1,\dots,n-1$, parce que les monômes a_i sont ordonnés lexicographiquement. De plus, comme les syzygies $S(a_i,a_{i+1})$ utilisent les éléments σ_i,σ_{i+1} de la base canonique, ils sont donc indépendants dans $\mathbb{K}[x,y]^n$. Par suite $\mathcal{L}(\mathbf{a})$ est un module libre de rang $n-1$ et donc on a la résolution suivante :

$$0\to\mathbb{K}[x,y]^{n-1}\to\mathbb{K}[x,y]^n\to(\mathbf{a})\to 0.$$

Supposons maintenant que les a_i sont des polynômes quelconques dans $\mathbb{K}[x,y]$. On va calculer la base de Gröbner de a_i, pour un ordre monomial qui raffine le degré (voir [37]).

On dénote par $m_1,\dots,m_s$ les termes principaux de ces polynômes dans cette base de Gröbner, ordonnée par l'ordre lexicographique.

Cette construction précédente permet d'obtenir la résolution de $(m_1,\dots,m_s)$ suivante :

$$0\to\mathbb{K}[x,y]^{s-1}\to\mathbb{K}[x,y]^s\to(m_i)_{i=1,\dots,s}\to 0.$$

En utilisant [84] (ou [37]), cette résolution peut être déformée à une résolution de $(\mathbf{a})$, de la forme :

$$0\to\mathbb{K}[x,y]^p\to\mathbb{K}[x,y]^n\to(\mathbf{a})\to 0,$$

pour un certain p. Ce qui montre que $\mathcal{L}(\mathbf{a})$ est aussi un module libre. Son rang p est obligatoirement égal $n-1$, parce que la somme alternée des dimensions des espaces vectoriels des éléments de degrés $\leq\nu$ dans chaque module de cette résolution, égale 0, pour $\nu\in\mathbb{N}$. □

On va, maintenant, construire une base de $\mathcal{L}(\mathbf{T})$.

Définition 6.3.11. *On note par $\{\sigma_1,\dots,\sigma_9\}$ la base canonique de $\mathbb{K}[x,y]$-module $\mathbb{K}[x,y]^9$.*

Le polynôme $\tilde{T}(x,y)$ est de degré $\leq 2m-1$ en x et $\leq 2n-1$ en y et comme la fonction (6.28) est surjective, alors il existe $u_1,u_2\in\mathbb{K}[x,y]^9_{\substack{m-1\\n-1}}$ tels que $\mathbf{T}\cdot u_1=\tilde{T}(x,y)x^m$ et $\mathbf{T}\cdot u_2=\tilde{T}(x,y)y^n$. C'est à dire,

$$\begin{aligned}\rho_1&=x^m\sigma_1-u_1\in\mathcal{L}(\mathbf{T}),\\ \rho_2&=y^n\sigma_1-u_2\in\mathcal{L}(\mathbf{T}).\end{aligned}$$

Pour les mêmes raisons, il existe $u_3\in\mathbb{K}[x,y]_{\substack{m-1\\n-1}}$, tel que $\mathbf{T}\cdot u_3=1=x^m\,x^m-(x^{2\,m}-1)=y^n\,y^n-(y^{2\,n}-1)$. On a donc

$$\begin{aligned}\rho_3&=x^m\sigma_2-\sigma_3-u_3\in\mathcal{L}(\mathbf{T}),\\ \rho_4&=y^n\sigma_4-\sigma_7-u_3\in\mathcal{L}(\mathbf{T}).\end{aligned}$$

De plus, on a les relations évidentes suivantes :

$$\begin{aligned}
\rho_5 &= y^n\sigma_2 - \sigma_5 \in \mathcal{L}(\mathbf{T}),\\
\rho_6 &= x^m\sigma_4 - \sigma_5 \in \mathcal{L}(\mathbf{T}),\\
\rho_7 &= x^m\sigma_5 - \sigma_6 + \sigma_4 \in \mathcal{L}(\mathbf{T}),\\
\rho_8 &= y^n\sigma_5 - \sigma_8 + \sigma_2 \in \mathcal{L}(\mathbf{T}).
\end{aligned}$$

Proposition 6.3.12. *Les relations $\rho_1, \ldots, \rho_8$ forment une base de $\mathcal{L}(\mathbf{T})$.*

Démonstration. Soit $\mathbf{h} = (h_1, \ldots, h_9) \in \mathcal{L}(\mathbf{T})$. En réduisant par $\rho_1, \ldots, \rho_8$, on peut admettre que les coefficients h_1, h_2, h_4, h_5 sont dans $\mathbb{K}[x,y]_{\substack{m-1\\n-1}}$. Donc, $\tilde{T}(x,y)h_1 + x^m h_2 + y^n h_4 + x^m y^n h_5 \in (x^{2\,n} - 1, y^{2\,m} - 1)$. Comme ce polynôme est de degré $\leq 3\,m - 1$ en x et $\leq 3\,n - 1$ en y, donc, par réduction par $\rho_1, \ldots, \rho_8$, on déduit que les coefficients $h_3, h_6, \ldots, h_9$ sont dans $\mathbb{K}[x,y]_{\substack{m-1\\n-1}}$. Par la proposition 6.3.8, il n'existe pas de syzygies non nul dans $\mathbb{K}[x,y]^9_{\substack{m-1\\n-1}}$. Donc, après réduction par $\rho_1, \ldots, \rho_8$, on a $\mathbf{h} = 0$. Donc, n'importe quel élément de $\mathcal{L}(\mathbf{T})$ peut être réduit à 0 par les relations précédentes. En d'autres mots, $\rho_1, \ldots, \rho_8$ engendrent le $\mathbb{K}[x,y]$-module $\mathcal{L}(\mathbf{T})$. Par le théorème 6.3.10, les relations ρ_i ne peuvent pas être dépendants sur $\mathbb{K}[x,y]$ et donc, elles forment une base de $\mathcal{L}(\mathbf{T})$. □

Proposition 6.3.13. *Le reste de la division de $(0,\, x^m g,\, g,\, 0,\, \ldots,\, 0)^T$ par $\{\rho_1, \ldots, \rho_8\}$ est l'unique vecteur de $\mathcal{L}(\mathbf{T}; g) \cap \mathbb{K}[x,y]^9_{\substack{m-1\\n-1}}$, ce qui donne la solution u.*

Démonstration. Posons $V = (0,\, x^m g,\, g,\, 0,\, \ldots,\, 0)^T$ et $\mathcal{L}(\mathbf{T}; g) \cap \mathbb{K}[x,y]^9_{\substack{m-1\\n-1}} = \{U\}$. On peut remarque facilement que $V \in \mathcal{L}(\mathbf{T}; g)$, donc $V - U \in \mathcal{L}(\mathbf{T})$. Comme $\{\rho_1, \ldots, \rho_8\}$ forme une base de $\mathcal{L}(\mathbf{T})$ alors il existe des uniques $p_1, \ldots, p_8$ tels que

$$V = \sum_{i=1}^{8} p_i\rho_i + U$$

□

6.3.3 Construction des générateurs et division

Après cette étude théorique qui nous permet de voir la solution du problème $Tu = g$ comme la reste d'une division par une base de $\mathcal{L}(\mathbf{T})$, on s'instéresse maintenant à la recherche des algorithmes qui calculent rapidement une base de $\mathcal{L}(\mathbf{T})$ puis faire rapidement la division.

On peut simplifier un peut le problème du calcul d'une base, parce que on est intéressé de calculer seulement les coefficients de $\sigma_1, \sigma_2, \sigma_4, \sigma_5$ de ρ_1, ρ_2, ρ_3. Notons $B(x,y)$ la matrice de coefficients correspondante, elle est donc de la forme suivante :

$$\begin{pmatrix} x^m & y^n & 0\\ 0 & 0 & x^m\\ 0 & 0 & 0\\ 0 & 0 & 0 \end{pmatrix} + \mathbb{K}[x,y]^{4,3}_{\substack{m-1\\n-1}} \tag{6.32}$$

Notons que les autres coefficients de relations ρ_1, ρ_2, ρ_3 correspondent aux éléments dans l'idéal $(x^{2\,m} - 1, y^{2\,n} - 1)$. Donc, on peut les obtenir en réduisant les coefficients de $(\tilde{T}(x, y), x^m, y^n, x^m\, y^n) \cdot B(x, y)$ par les polynômes $x^{2\,m} - 1, y^{2\,n} - 1$.

Notons aussi que la relation ρ_4 peut être déduite de ρ_3. En effet, ona $\rho_3 - x^m\sigma_2 + \sigma_3 + y^n\,\sigma_4 - \sigma_7 = \rho_4$. Comme les autres relations, ρ_i, $4 < i \leq 8$, sont données explicitement et elles sont indépendantes de $\tilde{T}(x, y)$, on peut déduire donc une base de $\mathcal{L}(\mathbf{T})$ à partir de la matrice $B(x, y)$.

Dans le cas scalaire on a expliqué comment on peut faire ce calcul rapidement. Dans le cas de Toeplitz par blocs de Toeplitz, le problème est beaucoup plus difficile et on ne dispose pas des algorithmes rapides qui calculent B ni des algorithmes rapides qui font la division. Cette difficulté vient de la difficulté du calcul des syzygies en deux variables, et de la difficulté de réduction en deux variables.

Conclusion et perspectives

Conclusion

Le but de cette thèse était de trouver un algorithme de résolution rapide des systèmes linéaires de Toeplitz par blocs de Toeplitz (TBT). On n'a pas pu aboutir à notre but, parce que ce problème s'est révélé énormement plus difficile qu'on pensait à priori. On a détaillé ces difficultés dans le quatrième chapitre, où on demande si un tel algorithme peut exister. On a essayé de résoudre ces systèmes en donnant des relations entre matrices TBT est polynômes. Cette méthode a bien marché dans le cas scalaire. La généralisation de cette méthode est possible pour le cas TBT, on obtient des relations importantes entre la solution d'un système TBT et les polynômes en deux variables, ces relations sont équivalentes à une formule de Gohberg-Semencul pour les matrices TBT. Dans le cas particulier des matrices Toeplitz bande par blocs Toeplitz bande, on a donné trois algorithmes rapides de résolution. Dans le deuxième chapitre on explique avec détailles et beaucoup de références ce qu'on sait faire dans le cas scalaire. Dans le troisième chapitre on donne beaucoup d'applications qui motivent ce sujet.

Perspectives

Après une longue étude de ce problème, qui est toujours irrésolu, je pense que ce qui peut ouvrir la porte à la résolution de ce problème est de créer une nouvelle théorie qui généralise la notion de la structure de déplacement. Cette généralisation n'est pas facile parce qu'il ne faut pas chercher à obtenir, en appliquant des opérateurs de déplacement, des matrices de petit rang ; mais par contre il faut chercher à généraliser la notion de petit rang qui ne reste pas vrai en deux niveaux. Sans développer cette théorie, il ne faut pas essayer de généraliser les algorithmes de cas scalaire, parce qu'ils ne marcheront pas dans le cas de deux niveaux ; il ne faut pas, non plus, essayer de chercher des opérateurs de déplacement, pour le cas de deux niveaux, qui vérifient les trois conditions d'un opérateur du cas scalaire.

La relation, qu'on donne dans le dernier chapitre, entre la solution d'un système TBT et syzygies est très importante. La continuation dans cette direction de recherche peut aboutir à des résultats importants. []

Bibliographie

[1] Hirotugu Akaike. Block Toeplitz matrix inversion. *SIAM J. Appl. Math.*, 24 :234–241, 1973.

[2] Gregory Ammar and Paul Gader. A variant of the Gohberg-Semencul formula involving circulant matrices. *SIAM J. Matrix Anal. Appl.*, 12(3) :534–540, 1991.

[3] Gregory S. Ammar. Classical foundations of algorithms for solving positive definite Toeplitz equations. *Calcolo*, 33(1-2) :99–113 (1998), 1996. Toeplitz matrices : structures, algorithms and applications (Cortona, 1996).

[4] Gregory S. Ammar and William B. Gragg. The generalized Schur algorithm for the superfast solution of Toeplitz systems. In *Rational approximation and applications in mathematics and physics (Łańcut, 1985)*, volume 1237 of *Lecture Notes in Math.*, pages 315–330. Springer, Berlin, 1987.

[5] Gregory S. Ammar and William B. Gragg. Superfast solution of real positive definite Toeplitz systems. In *Linear algebra in signals, systems, and control (Boston, MA, 1986)*, pages 107–125. SIAM, Philadelphia, PA, 1988.

[6] Greg W. Anderson and Ofer Zeitouni. A CLT for a band matrix model. *Probab. Theory Related Fields*, 134(2) :283–338, 2006.

[7] R. H. Bartels and G. W. Stewart. Solution of the matrix equation ax + xb = c [f4]. *Commun. ACM*, 15(9) :820–826, 1972.

[8] Dario Bini and Victor Pan. Improved parallel computations with Toeplitz-like and Hankel-like matrices. *Linear Algebra Appl.*, 188/189 :3–29, 1993.

[9] Dario Bini and Victor Y. Pan. *Polynomial and matrix computations. Vol. 1.* Progress in Theoretical Computer Science. Birkhäuser Boston Inc., Boston, MA, 1994. Fundamental algorithms.

[10] Dario Andrea Bini and Beatrice Meini. Approximate displacement rank and applications. In *Structured matrices in mathematics, computer science, and engineering, II (Boulder, CO, 1999)*, volume 281 of *Contemp. Math.*, pages 215–232. Amer. Math. Soc., Providence, RI, 2001.

[11] R.R. Bitmead and B.D.O. Anderson. Asymptotically fast solution of Toeplitz and related systems of equations. 34 :103–116, 1980.

[12] Simon R. Blackburn. Fast rational interpolation, Reed-Solomon decoding, and the linear complexity profiles of sequences. *IEEE Trans. Inform. Theory*, 43(2) :537–548, 1997.

[13] D. Bondyfalat, B. Mourrain, and V. Y. Pan. Controlled iterative methods for solving polynomial systems. In O. Gloor, editor, *Proc. Intern. Symp. on Symbolic and Algebraic Computation*, pages 252–259. NewYork, ACM Press., 1998.

[14] Arup Bose and Arnab Sen. Spectral norm of random large dimensional noncentral Toeplitz and Hankel matrices. *Electron. Comm. Probab.*, 12 :29–35 (electronic), 2007.

[15] Alin Bostan, Claude-Pierre Jeannerod, and Éric Schost. Solving toeplitz- and vandermonde-like linear systems with large displacement rank. In *ISSAC '07 : Proceedings of the 2007 international symposium on Symbolic and algebraic computation*, pages 33–40, New York, NY, USA, 2007. ACM.

[16] Richard P. Brent, Fred G. Gustavson, and David Y. Y. Yun. Fast solution of Toeplitz systems of equations and computation of Padé approximants. *J. Algorithms*, 1(3) :259–295, 1980.

[17] W. S. Brown. On Euclid's algorithm and the computation of polynomial greatest common divisors. *J. Assoc. Comput. Mach.*, 18 :478–504, 1971.

[18] James R. Bunch. Stability of methods for solving Toeplitz systems of equations. *SIAM J. Sci. Statist. Comput.*, 6(2) :349–364, 1985.

[19] Laurent Busé, Houssam Khalil, and Bernard Mourrain. Resultant-based methods for plane curves intersection problems. In *Computer algebra in scientific computing*, volume 3718 of *Lecture Notes in Comput. Sci.*, pages 75–92. Springer, Berlin, 2005.

[20] John Canny. *The complexity of robot motion planning*, volume 1987 of *ACM Doctoral Dissertation Awards*. MIT Press, Cambridge, MA, 1988.

[21] Raymond H. Chan and Xiao-Qing Jin. A family of block preconditioners for block systems. *SIAM J. Sci. Statist. Comput.*, 13(5) :1218–1235, 1992.

[22] Raymond H. Chan and Michael K. Ng. Conjugate gradient methods for Toeplitz systems. *SIAM Rev.*, 38(3) :427–482, 1996.

[23] Tony F. Chan and Hansen. A look-ahead levinson algorithm for general toeplitz systems systems. *IEEE Trans. Signal Process.*, 40 :1079–1090, 1992.

[24] Tony F. Chan and Per Christian Hansen. A look-ahead Levinson algorithm for indefinite Toeplitz systems. *SIAM J. Matrix Anal. Appl.*, 13(2) :490–506, 1992.

[25] Falai Chen, David Cox, and Yang Liu. The μ-basis and implicitization of a rational parametric surface. *J. Symbolic Comput.*, 39(6) :689–706, 2005.

[26] J. Chun and T. Kailath. A constructive proof of the Gohberg-Semencul formula. *Linear Algebra Appl.*, 121 :475–489, 1989. Linear algebra and applications (Valencia, 1987).

[27] P.G Ciarlet. *Handbook of numerical analysis : finite element methods*. North-Holland, 1990.

[28] D. Commenges and M. Monsion. Fast inversion of triangular Toeplitz matrices. *IEEE Trans. Automat. Control*, 29(3) :250–251, 1984.

[29] David Cox, John Little, and Donal O'Shea. *Using algebraic geometry*, volume 185 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, 1998.

[30] David A. Cox. Equations of parametric curves and surfaces via syzygies. In *Symbolic computation : solving equations in algebra, geometry, and engineering (South Hadley, MA, 2000)*, volume 286 of *Contemp. Math.*, pages 1–20. Amer. Math. Soc., Providence, RI, 2001.

[31] David A. Cox, Thomas W. Sederberg, and Falai Chen. The moving line ideal basis of planar rational curves. *Comput. Aided Geom. Design*, 15(8) :803–827, 1998.

[32] G. Cybenko and M. Berry. Hyperbolic Householder algorithms for factoring structured matrices. *SIAM J. Matrix Anal. Appl.*, 11(4) :499–520, 1990.

[33] Frank de Hoog. A new algorithm for solving Toeplitz systems of equations. *Linear Algebra Appl.*, 88/89 :123–138, 1987.

[34] Philippe Delsarte and Yves V. Genin. The split Levinson algorithm. *IEEE Trans. Acoust. Speech Signal Process.*, 34(3) :470–478, 1986.

[35] Fabio Di Benedetto. Preconditioning of block Toeplitz matrices by sine transforms. *SIAM J. Sci. Comput.*, 18(2) :499–515, 1997.

[36] Ömer Eğecioğlu and Çetin K. Koç. A fast algorithm for rational interpolation via orthogonal polynomials. *Math. Comp.*, 53(187) :249–264, 1989.

[37] David Eisenbud. *Commutative algebra*, volume 150 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, 1995. With a view toward algebraic geometry.

[38] I.Z. Emiris and Mourrain Bernard. Polynomial system solving ; the case of a 6-atom molecule. rapport de recherche 3075. INRIA, 1996.

[39] I.Z. Emiris and B. Mourrain. Matrices in Elimination Theory. *J. of Symbolic Computation*, 28(1&2) :3–44, 1999.

[40] Alexandre Ern and Jean-Luc Guermond. *Theory and practice of finite elements*, volume 159 of *Applied Mathematical Sciences*. Springer-Verlag, New York, 2004.

[41] Tilo Finck, Georg Heinig, and Karla Rost. An inversion formula and fast algorithms for Cauchy-Vandermonde matrices. *Linear Algebra Appl.*, 183 :179–191, 1993.

[42] Giuseppe Fiorentino and Stefano Serra. Multigrid methods for symmetric positive definite block Toeplitz matrices with nonnegative generating functions. *SIAM J. Sci. Comput.*, 17(5) :1068–1081, 1996.

[43] B. Friedlander, M. Morf, T. Kailath, and L. Ljung. New inversion formulas for matrices classified in terms of their distance from Toeplitz matrices. *Linear Algebra Appl.*, 27 :31–60, 1979.

[44] P.A. Fuhrmann. *A polynomial approach to linear algebra*. Springer-Verlag, 1996.

[45] K. A. Gallivan, S. Thirumalai, P. Van Dooren, and V. Vermaut. High performance algorithms for Toeplitz and block Toeplitz matrices. In *Proceedings of the Fourth Conference of the International Linear Algebra Society (Rotterdam, 1994)*, volume 241/243, pages 343–388, 1996.

[46] Luca Gemignani. Schur complements of Bezoutians and the inversion of block Hankel and block Toeplitz matrices. *Linear Algebra Appl.*, 253 :39–59, 1997.

[47] A. Gerasoulis. A fast algorithm for the multiplication of generalized Hilbert matrices with vectors. *Math. Comp.*, 50(181) :179–188, 1988.

[48] I. Gohberg, T. Kailath, and I. Koltracht. Efficient solution of linear systems of equations with recursive structure. *Linear Algebra Appl.*, 80 :81–113, 1986.

[49] I. Gohberg, T. Kailath, I. Koltracht, and P. Lancaster. Linear complexity parallel algorithms for linear systems of equations with recursive structure. *Linear Algebra Appl.*, 88/89 :271–315, 1987.

[50] I. Gohberg, T. Kailath, and V. Olshevsky. Fast Gaussian elimination with partial pivoting for matrices with displacement structure. *Math. Comp.*, 64(212) :1557–1576, 1995.

[51] I. Gohberg and V. Olshevsky. Circulants, displacements and decompositions of matrices. *Integral Equations Operator Theory*, 15(5) :730–743, 1992.

[52] I. Gohberg and V. Olshevsky. Complexity of multiplication with vectors for structured matrices. *Linear Algebra Appl.*, 202 :163–192, 1994.

[53] I. C. Gohberg and A. A. Semencul. The inversion of finite Toeplitz matrices and their continual analogues. *Mat. Issled.*, 7(2(24)) :201–223, 290, 1972.

[54] Alexander Graham. *Kronecker products and matrix calculus : with applications*. Ellis Horwood Ltd., Chichester, 1981. Ellis Horwood Series in Mathematics and its Applications.

[55] D. Yu. Grigor′ev and N. N. Vorobjov, Jr. Solving systems of polynomial inequalities in subexponential time. *J. Symbolic Comput.*, 5(1-2) :37–64, 1988.

[56] Philippe Guillaume. Nested multivariate Padé approximants. *J. Comput. Appl. Math.*, 82(1-2) :149–158, 1997. 7th ICCAM 96 Congress (Leuven).

[57] Philippe Guillaume. Convergence of the nested multivariate Padé approximants. *J. Approx. Theory*, 94(3) :455–466, 1998.

[58] Philippe Guillaume and Alain Huard. Multivariate Padé approximation. *J. Comput. Appl. Math.*, 121(1-2) :197–219, 2000. Numerical analysis in the 20th century, Vol. I, Approximation theory.

[59] Philippe Guillaume, Alain Huard, and Vincent Robin. Generalized multivariate Padé approximants. *J. Approx. Theory*, 95(2) :203–214, 1998.

[60] Alice Guionnet. Large deviations and stochastic calculus for large random matrices. *Probab. Surv.*, 1 :72–172 (electronic), 2004.

[61] Martin H. Gutknecht and Marlis Hochbruck. Look-ahead Levinson and Schur algorithms for non-Hermitian Toeplitz systems. *Numer. Math.*, 70(2) :181–227, 1995.

[62] M Heideman, D Johnson, and C. Burrus. Gauss and the history of the fast fourier transform. *IEEE ASSP Magazine*, 1 :14–21, 1984.

[63] Georg Heinig. Solving Toeplitz systems after extension and transformation. *Calcolo*, 33(1-2) :115–129 (1998), 1996. Toeplitz matrices : structures, algorithms and applications (Cortona, 1996).

[64] Georg Heinig and Karla Rost. *Algebraic methods for Toeplitz-like matrices and operators*, volume 13 of *Operator Theory : Advances and Applications*. Birkhäuser Verlag, Basel, 1984.

[65] Thomas Huckle. Computations with Gohberg-Semencul-type formulas for Toeplitz matrices. *Linear Algebra Appl.*, 273 :169–198, 1998.

[66] A. Jain. Fast inversion of banded toeplitz matrices by circular decompositions. *Acoustics, Speech, and Signal Processing, IEEE Transactions on*, 26(1-4) :121 – 126, Apr 1978.

[67] T. Kailath and J. Chun. Generalized Gohberg-Semencul formulas for matrix inversion. In *The Gohberg anniversary collection, Vol. I (Calgary, AB, 1988)*, volume 40 of *Oper. Theory Adv. Appl.*, pages 231–246. Birkhäuser, Basel, 1989.

[68] T. Kailath and A. H. Sayed, editors. *Fast reliable algorithms for matrices with structure*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1999.

[69] Thomas Kailath. Signal processing applications of some moment problems. In *Moments in mathematics (San Antonio, Tex., 1987)*, volume 37 of *Proc. Sympos. Appl. Math.*, pages 71–109. Amer. Math. Soc., Providence, RI, 1987.

[70] Thomas Kailath, Sun Yuan Kung, and Martin Morf. Displacement ranks of matrices and linear equations. *J. Math. Anal. Appl.*, 68(2) :395–407, 1979.

[71] Thomas Kailath and Ali H. Sayed. Displacement structure : theory and applications. *SIAM Rev.*, 37(3) :297–386, 1995.

[72] E Kaltofen. Asymptotically fast solution of toeplitz-like singular linear systems. In *ISSAC '94 : Proceedings of the international symposium on Symbolic and algebraic computation*, pages 297–304, New York, NY, USA, 1994. ACM Press.

[73] Erich Kaltofen. Analysis of Coppersmith's block Wiedemann algorithm for the parallel solution of sparse linear systems. *Math. Comp.*, 64(210) :777–806, 1995.

[74] A. Khorunzhy. Sparse random matrices : spectral edge and statistics of rooted trees. *Adv. in Appl. Probab.*, 33(1) :124–140, 2001.

[75] A. Khorunzhy and W. Kirsch. On asymptotic expansions and scales of spectral universality in band random matrix ensembles. *Comm. Math. Phys.*, 231(2) :223–255, 2002.

[76] M. Kin-Wai and A.H. Chan. Parallel implementation of 2-dimensional toeplitz solver on masparwith applications to image restoration. In *High performance computing on the information superhighway, 1997. HPC Asia '97*, pages 389–394, Seoul, South Korea, 1997.

[77] Peter Kravanja and Marc Van Barel. A fast Hankel solver based on an inversion formula for Loewner matrices. *Linear Algebra Appl.*, 282(1-3) :275–295, 1998.

[78] George Labahn and Tamir Shalom. Inversion of Toeplitz matrices with only two standard equations. *Linear Algebra Appl.*, 175 :143–158, 1992.

[79] Alessandro Logar and Bernd Sturmfels. Algorithms for the Quillen-Suslin theorem. *J. Algebra*, 145(1) :231–239, 1992.

[80] M. Mahboub and B. Philippe. Parallélisation du ré-échantillonnage d'images. In *CARI*, 2004.

[81] Mark W. Meckes. On the spectral norm of a random Toeplitz matrix. *Electron. Comm. Probab.*, 12 :315–325 (electronic), 2007.

[82] Gulamabbas A. Merchant and Thomas W. Parks. Efficient solution of a Toeplitz-plus-Hankel coefficient matrix system of equations. *IEEE Trans. Acoust. Speech Signal Process.*, 30(1) :40–44, 1982.

[83] S. A. Molchanov, L. A. Pastur, and A. M. Khorunzhiĭ. Distribution of the eigenvalues of random band matrices in the limit of their infinite order. *Teoret. Mat. Fiz.*, 90(2) :163–178, 1992.

[84] H. Michael Möller and Ferdinando Mora. New constructive methods in classical ideal theory. *J. Algebra*, 100(1) :138–178, 1986.

[85] M. Morf. *Fast algorithms for multivariable systems*. Ph.D thesis. Departement of Electrical Engineering, Stanford University, Stanford, CA, 1974.

[86] M Morf. Doubling algorithms for toeplitz and related equations. *Proc. IEEE Intern. Conf. on ASSP*, 5 :954–959, 1980.

[87] B. Mourrain and V. Y. Pan. Solving special polynomial systems by using structured matrices and algebraic residues. In F. Cucker and M. Shub, editors, *Foundations of Computational Mathematics (Rio de Janeiro)*, pages 287–304. Springer-Verlag, 1997.

[88] Bernard Mourrain and Victor Y. Pan. Multivariate polynomials, duality, and structured matrices. *J. Complexity*, 16(1) :110–180, 2000. Real computation and complexity (Schloss Dagstuhl, 1998).

[89] Bruce Musicus. Levinson and fast choleski algorithms for toeplitz and almost toeplitz matrices. *RLE technical report No. 538*, 1988.

[90] J.C Nédélec. *Notions sur les techniques d'éléments finis*. Ellipses, Paris, 1991.

[91] Michael K. Ng. *Iterative methods for Toeplitz systems*. Numerical Mathematics and Scientific Computation. Oxford University Press, New York, 2004.

[92] Michael K. Ng, Karla Rost, and You-Wei Wen. On inversion of Toeplitz matrices. *Linear Algebra Appl.*, 348 :145–151, 2002.

[93] D. Noutsos, S. Serra Capizzano, and P. Vassalos. Spectral equivalence and matrix algebra preconditioners for multilevel Toeplitz systems : a negative result. In *Fast algorithms for structured matrices : theory and applications (South Hadley, MA, 2001)*, volume 323 of *Contemp. Math.*, pages 313–322. Amer. Math. Soc., Providence, RI, 2003.

[94] S. T. O'Donnell and V. Rokhlin. A fast algorithm for the numerical evaluation of conformal mappings. *SIAM J. Sci. Statist. Comput.*, 10(3) :475–487, 1989.

[95] Vadim Olshevsky. Pivoting for structured matrices and rational tangential interpolation. In *Fast algorithms for structured matrices : theory and applications (South Hadley, MA, 2001)*, volume 323 of *Contemp. Math.*, pages 1–73. Amer. Math. Soc., Providence, RI, 2003.

[96] Vadim Olshevsky, Ivan Oseledets, and Eugene Tyrtyshnikov. Tensor properties of multilevel Toeplitz and related matrices. *Linear Algebra Appl.*, 412(1) :1–21, 2006.

[97] Vadim Olshevsky and Victor Pan. A unified superfast algorithm for boundary rational tangential interpolation problems and for inversion and factorization of dense structured matrices. *focs*, 00 :192, 1998.

[98] V. Y. Pan, M. A. Tabanjeh, Z. Q. Chen, E. I. Landowne, and A. Sadikou. New transformations of Cauchy matrices and Trummer's problem. *Comput. Math. Appl.*, 35(12) :1–5, 1998.

[99] Victor Pan. On computations with dense structured matrices. *Math. Comp.*, 55(191) :179–190, 1990.

[100] Victor Pan. Parallel solution of Toeplitzlike linear systems. *J. Complexity*, 8(1) :1–21, 1992.

[101] Victor Pan. Parametrization of Newton's iteration for computations with structured matrices and applications. *Comput. Math. Appl.*, 24(3) :61–75, 1992.

[102] Victor Pan. Decreasing the displacement rank of a matrix. *SIAM J. Matrix Anal. Appl.*, 14(1) :118–121, 1993.

[103] Victor Pan, Akimou Sadikou, Elliott Landowne, and Olen Tiga. A new approach to fast polynomial interpolation and multipoint evaluation. *Comput. Math. Appl.*, 25(9) :25–30, 1993.

[104] Victor Y. Pan. Nearly optimal computations with structured matrices. In *Proceedings of the Eleventh Annual ACM-SIAM Symposium on Discrete Algorithms (San Francisco, CA, 2000)*, pages 953–962, New York, 2000. ACM.

[105] Victor Y. Pan. *Structured matrices and polynomials*. Birkhäuser Boston Inc., Boston, MA, 2001. Unified superfast algorithms.

[106] Victor Y. Pan, Sheryl Branham, Rhys E. Rosholt, and Ai-Long Zheng. Newton's iteration for structured matrices. In *Fast reliable algorithms for matrices with structure*, pages 189–210. SIAM, Philadelphia, PA, 1999.

[107] Victor Y. Pan, Youssef Rami, and Xinmao Wang. Structured matrices and newton's iteration : unified approach. *Linear Algebra Appl.*, 343/344 :233–265, 2002. Special issue on structured and infinite systems of linear equations.

[108] Victor Y. Pan and Ailong Zheng. Superfast algorithms for Cauchy-like matrix computations and extensions. *Linear Algebra Appl.*, 310(1-3) :83–108, 2000.

[109] M. Raghavan and B. Roth. Solving polynomial systems for the kinematic analysis and synthesis of mechanisms and robot manipulators. *Journal of Vibration and Acoustics*, 117(B) :71–79, 1995.

[110] Pierre-Arnaud Raviart and Jean-Marie Thomas. *Introduction à l'analyse numérique des équations aux dérivées partielles*. Masson, Paris, 1993.

[111] V. Rokhlin. Rapid solution of integral equations of classical potential theory. *J. Comput. Phys.*, 60(2) :187–207, 1985.

[112] Lionel Sainsaulieu. *Calcul scientifique*. DUNOD, Paris, 2000.

[113] A. H. Sayed, T. Kailath, H. Lev-Ari, and T. Constantinescu. Recursive solutions of rational interpolation problems via fast matrix factorization. *Integral Equations Operator Theory*, 20(1) :84–118, 1994.

[114] Micchelle Schatzman. Communication personnelle.

[115] Michelle Schatzman. *Analyse numérique*. InterEditions, Paris, 1991. Cours et exercices pour la licence. [Course and exercises for the bachelor's degree].

[116] S. Serra Capizzano and E. Tyrtyshnikov. Any circulant-like preconditioner for multilevel matrices is not superlinear. *SIAM J. Matrix Anal. Appl.*, 21(2) :431–439 (electronic), 1999.

[117] S. Serra Capizzano and E. Tyrtyshnikov. How to prove that a preconditioner cannot be superlinear. *Math. Comp.*, 72(243) :1305–1316 (electronic), 2003.

[118] Michael Stewart. A superfast Toeplitz solver with improved numerical stability. *SIAM J. Matrix Anal. Appl.*, 25(3) :669–693 (electronic), 2003.

[119] William F. Trench. An algorithm for the inversion of finite Toeplitz matrices. *J. Soc. Indust. Appl. Math.*, 12 :515–522, 1964.

[120] William Turin. *Performance analysis of digital transmission systems*. Computer Science Press, Inc., New York, NY, USA, 1990.

[121] E.E. Tyrtyshnikov. Fast algorithms for block Toeplitz matrices. *Sov. J. Numer. Math. Modelling*, 1(2) :121–139, 1985.

[122] M. Van Barel and A. Bultheel. A general module-theoretic framework for vector M-Padé and matrix rational interpolation. *Numer. Algorithms*, 3(1-4) :451–461, 1992. Extrapolation and rational approximation (Puerto de la Cruz, 1992).

[123] M. Van Barel and Z. Vavřín. Inversion of a block Löwner matrix. *J. Comput. Appl. Math.*, 69(2) :261–284, 1996.

[124] Marc Van Barel and Adhemar Bultheel. A new approach to the rational interpolation problem. *J. Comput. Appl. Math.*, 32(1-2) :281–289, 1990. Extrapolation and rational approximation (Luminy, 1989).

[125] Marc Van Barel and Adhemar Bultheel. A new approach to the rational interpolation problem : the vector case. *J. Comput. Appl. Math.*, 33(3) :331–346, 1990.

[126] Marc Van Barel and Adhemar Bultheel. A look-ahead method for computing vector Padé-Hermite approximants. *Constr. Approx.*, 11(4) :455–476, 1995.

[127] Marc Van Barel and Adhemar Bultheel. Look-ahead methods for block Hankel systems. *J. Comput. Appl. Math.*, 86(1) :311–333, 1997. Special issue dedicated to William B. Gragg (Monterey, CA, 1996).

[128] Marc Van Barel and Adhemar Bultheel. A lookahead algorithm for the solution of block Toeplitz systems. *Linear Algebra Appl.*, 266 :291–335, 1997.

[129] Marc Van Barel, Georg Heinig, and Peter Kravanja. A stabilized superfast solver for nonsymmetric Toeplitz systems. *SIAM J. Matrix Anal. Appl.*, 23(2) :494–510 (electronic), 2001.

[130] Marc Van Barel and Peter Kravanja. A stabilized superfast solver for indefinite Hankel systems. *Linear Algebra Appl.*, 284(1-3) :335–355, 1998. ILAS Symposium on Fast Algorithms for Control, Signals and Image Processing (Winnipeg, MB, 1997).

[131] Marc Van Barel and Peter Kravanja. On the generically superfast computation of Hankel determinants. In *Large-scale scientific computations of engineering and environmental problems, II (Sozopol, 1999)*, volume 73 of *Notes Numer. Fluid Mech.*, pages 57–64. Vieweg, Braunschweig, 2000.

[132] Charles Van Loan. *Computational frameworks for the fast Fourier transform*, volume 10 of *Frontiers in Applied Mathematics*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1992.

[133] Joachim von zur Gathen. Parallel algorithms for algebraic problems. *SIAM J. Comput.*, 13(4) :802–824, 1984.

[134] Joachim von zur Gathen and Jürgen Gerhard. *Modern computer algebra*. Cambridge University Press, Cambridge, second edition, 2003.

[135] Mati Wax and Thomas Kailath. Efficient inversion of Toeplitz-block Toeplitz matrix. *IEEE Trans. Acoust. Speech Signal Process.*, 31(5) :1218–1221, 1983.

[136] D. Wiedemann. Solving sparse linear equations over finite fields. *Information Theory, IEEE Transactions on*, 32(1) :54–62, Jan 1986.

[137] H. Yunbiao Wang Krishna. On fast and superfast algorithms for solving block toeplitz systems. *Signals, Systems and Computers, 1989. Twenty-Third Asilomar Conference on*, 2 :643 – 647, Oct. 30-Nov. 1, 1989.