ALGORITMOS EN ÁLGEBRA LINEAL

Notas de curso (UBA - 2do cuatrimestre de 2005)

http://atlas.mat.ub.es/personals/sombra/curso.html

Michelle Schatzman

Martín Sombra

Institut Camille Jordan (Mathématiques), Université de Lyon 1 ; 43 Bd. du 11 Novembre 1918, 69622 Villeurbanne Cedex, Francia

Universitat de Barcelona, Departament d'Àlgebra i Geometria ; Gran Via 585, 08007 Barcelona, España

CHAPITRE 5

Structure de déplacement

Nombre d'applications dans les sciences et les ingénieries font appel à des matrices structurées telles que les matrices circulantes, résultantes, Bézout, Toeplitz, Hankel, Vandermonde, Cauchy, Loewner, et Pick. Ces applications demandent souvent la résolution de systèmes linéaires de grande taille, d'où l'intéret pour les algorithmes de résolution adaptés à ces structures.

Les types de matrices mentionnées sont denses, or ses coefficients dépendent de $O(N)$ paramètres seulement. La notion de rang de déplacement permets d'unifier l'ensemble de ces structures dans une seule approche : toutes les matrices mentionnées ont un rang de déplacement $O(1)$ par rapport à des opérateurs convénables.

La notion de rang de déplacement est la clé pour l'obtention d'algorithmes efficaces pour des matrices structurées, pour des tâches comme la résolution de systèmes d'équations, calcul de noyau et d'image, inversion, multiplication matrice-vecteur, etc. Pour une matrice $A \in \mathbb{K}^{N\times N}$ au rang de déplacement α, on peut résoudre $Ax = b$ avec des algorithmes "rapides" en

$$O(\alpha N^2) \quad \text{ops}$$

et avec des algorithmes "super-rapides" en

$$O(\alpha^2 N \log^2(N)) \quad \text{ou} \quad O(\alpha^2 N \log^3(N)) \quad \text{ops}\,.$$

Plus important encore, cette approche permets non seulement de résoudre les structures classiques, mais aussi de traiter des matrices proches au sens qu'elles ont un petit rang de déplacement par rapport aux opérateurs associés à chacune de ces structures.

Le rang de déplacement d'une matrice fut introduit dans l'article [**9**] comme une mesure de sa proximité à la classe des matrices Toeplitz, *voir* aussi [**3**] ; cependant cette idée fut beaucoup plus profonde, puissante et générale qu'imaginé dans un premier moment. Parmi les antécedents, remarquons la thèse de Morf [**11**] et la celèbre formule de Gohberg et Semencul pour l'inverse d'une matrice de Toeplitz [**6**]. L'idée de déplacement fut étendue à des autres structures dans les articles [**8, 4, 5**].

Dans ce chapitre on introduira les bases de cette théorie et on illustrera les algorithmes avec una application importante, la décodification des codes correcteurs d'erreurs. Les principales références pour ce chapitre sont [**10, 12, 14**].

0.1. Rang de déplacement.

Définition 0.1. Soient $M, N \in \mathbb{N}$ des entier et $S \in \mathbb{K}^{M\times M}$ et $T \in \mathbb{K}^{N\times N}$ des matrices fixées, l'*opérateur de déplacement* associé est

$$\nabla_{S,T} : \mathbb{K}^{M\times N} \to \mathbb{K}^{M\times N} \quad , \quad A \mapsto S \cdot A - A \cdot T.$$

Le *rang de déplacement* ou (S,T)-*rang* ou encore ∇-*rang* est

$$\operatorname{rang}_{S,T}(A) := \operatorname{rang}(\nabla_{S,T}(A)).$$

On dira que A est (S,T)-*structurée* si

$$\operatorname{rang}_{S,T}(A) \ll \min(M,N);$$

bien entendu il s'agit d'une notion quantitative et non qualitative. Pour gagner en simplicité, on se restreindra au cas d'une matrice carré et inversible $A \in \mathbb{K}^{N\times N}$.

Pour $\lambda \in \mathbb{K}$ notons

$$Z_\lambda = \begin{bmatrix} 0 & 1 & & \\ & \ddots & \ddots & \\ & & \ddots & 1 \\ \lambda & & & 0 \end{bmatrix} \in \mathbb{K}^{N\times N},$$

en particulier Z_0 est le bloc de Jordan ou opérateur de shift de taille N. Passons revue aux quattre cas les plus celèbres de matrices structurées :

(1) Matrices de Toeplitz :

$$\mathcal{T} = [a_{i-j}]_{1\le i,j\le N} = \begin{bmatrix} a_0 & a_{-1} & \cdots & a_{-(N-1)} \\ a_1 & a_0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & a_{-1} \\ a_{(N-1)} & \cdots & a_1 & a_0 \end{bmatrix} \in \mathbb{K}^{N\times N}.$$

Les déplacement associés sont

$$S = Z_1 \quad , \quad T = Z_0.$$

Calculons le déplacement pour $N = 3$:

$$\begin{aligned} \nabla_{Z_1,Z_0}(\mathcal{T}) &= \begin{bmatrix} & 1 & \\ & & 1 \\ 1 & & \end{bmatrix} \cdot \begin{bmatrix} a_0 & a_{-1} & a_{-2} \\ a_1 & a_0 & a_{-1} \\ a_2 & a_1 & a_0 \end{bmatrix} - \begin{bmatrix} a_0 & a_{-1} & a_{-2} \\ a_1 & a_0 & a_{-1} \\ a_2 & a_1 & a_0 \end{bmatrix} \cdot \begin{bmatrix} & 1 & \\ & & 1 \\ 0 & & \end{bmatrix} \\ &= \begin{bmatrix} a_1 & a_0 & a_{-1} \\ a_2 & a_1 & a_0 \\ a_0 & a_{-1} & a_{-2} \end{bmatrix} - \begin{bmatrix} 0 & a_0 & a_{-1} \\ 0 & a_1 & a_0 \\ 0 & a_2 & a_1 \end{bmatrix} = \begin{bmatrix} a_1 & & \\ a_2 & & \\ a_0 & a_{-1}-a_2 & a_{-2}-a_1 \end{bmatrix}. \end{aligned}$$

Ce calcul est tout à fait général ; pour tout N on a $\operatorname{rang}_{Z_1,Z_0}(\mathcal{T}) \le 2$.

(2) Matrices de Hankel :

$$\mathcal{H} = [a_{i+j-2}]_{1\le i,j\le N} = \begin{bmatrix} a_0 & a_1 & \cdots & a_{N-1} \\ a_1 & a_2 & \diagup & a_N \\ \vdots & \diagup & \diagup & \vdots \\ a_{N-1} & a_N & \cdots & a_{2N-2} \end{bmatrix} \in \mathbb{K}^{N\times N}.$$

Dans ce cas on prends

$$S = Z_1 \quad , \quad T = Z_0^T,$$

faisons le calcul pour $N = 3$:

$$\begin{aligned} \nabla_{Z_1,Z_0^T}(\mathcal{H}) &= \begin{bmatrix} & 1 & \\ & & 1 \\ 1 & & \end{bmatrix} \cdot \begin{bmatrix} a_0 & a_1 & a_2 \\ a_1 & a_2 & a_3 \\ a_2 & a_3 & a_4 \end{bmatrix} - \begin{bmatrix} a_0 & a_1 & a_2 \\ a_1 & a_2 & a_3 \\ a_2 & a_3 & a_4 \end{bmatrix} \cdot \begin{bmatrix} & & 0 \\ 1 & & \\ & 1 & \end{bmatrix} \\ &= \begin{bmatrix} a_1 & a_2 & a_3 \\ a_2 & a_3 & a_4 \\ a_0 & a_1 & a_2 \end{bmatrix} - \begin{bmatrix} a_1 & a_2 & 0 \\ a_2 & a_3 & 0 \\ a_3 & a_4 & 0 \end{bmatrix} = \begin{bmatrix} & & a_3 \\ & & a_4 \\ a_0-a_3 & a_1-a_4 & a_2 \end{bmatrix}. \end{aligned}$$

Ce calcul est général, pour tout N on a $\operatorname{rang}_{Z_1,Z_0^T}(\mathcal{H}) \le 2$.

(3) Matrices de Vandermonde : pour $x_1, \dots, x_N \in \mathbb{K}^\times$

$$\mathcal{V}(x) = [x_i^{j-1}]_{1\le i,j\le N} = \begin{bmatrix} 1 & x_1 & \cdots & x_1^{N-1} \\ 1 & x_2 & \cdots & x_2^{N-1} \\ \vdots & \vdots & & \vdots \\ 1 & x_N & \cdots & x_N^{N-1} \end{bmatrix} \in \mathbb{K}^{N\times N}.$$

Pour les déplacements

$$S = \operatorname{diag}(x_1^{-1}, \dots, x_n^{-1}) \quad , \quad T = Z_0$$

on a $\operatorname{rang}_{S,T}(\mathcal{V}) = 1$. Vérifions cela pour $N = 3$:

$$\begin{aligned} \nabla_{S,T}(\mathcal{V}(x)) &= \begin{bmatrix} x_1^{-1} & & \\ & x_2^{-1} & \\ & & x_3^{-1} \end{bmatrix} \cdot \begin{bmatrix} 1 & x_1 & x_1^2 \\ 1 & x_2 & x_2^2 \\ 1 & x_3 & x_3^2 \end{bmatrix} - \begin{bmatrix} 1 & x_1 & x_1^2 \\ 1 & x_2 & x_2^2 \\ 1 & x_3 & x_3^2 \end{bmatrix} \cdot \begin{bmatrix} & 1 & \\ & & 1 \\ 0 & & \end{bmatrix} \\ &= \begin{bmatrix} x_1^{-1} & 1 & x_1 \\ x_2^{-1} & 1 & x_2 \\ x_3^{-1} & 1 & x_3 \end{bmatrix} - \begin{bmatrix} 0 & 1 & x_1 \\ 0 & 1 & x_2 \\ 0 & 1 & x_3 \end{bmatrix} = \begin{bmatrix} x_1^{-1} & 0 & 0 \\ x_2^{-1} & 0 & 0 \\ x_3^{-1} & 0 & 0 \end{bmatrix}. \end{aligned}$$

(4) Matrices de Cauchy : soient $x_1, \dots, x_N, y_1, \dots, y_N \in \mathbb{K}$ tels que $x_i \neq y_j$ pour tout $1 \le i, j \le N$, alors

$$\mathcal{C}(x,y) = [(x_i - y_j)^{-1}]_{1\le i,j\le N} = \begin{bmatrix} \frac{1}{x_1-y_1} & \frac{1}{x_1-y_2} & \cdots & \frac{1}{x_1-y_n} \\ \frac{1}{x_2-y_1} & \frac{1}{x_2-y_2} & \cdots & \frac{1}{x_2-y_N} \\ \vdots & \vdots & & \vdots \\ \frac{1}{x_N-y_1} & \frac{1}{x_N-y_2} & \cdots & \frac{1}{x_N-y_N} \end{bmatrix} \in \mathbb{K}^{N\times N}.$$

Dans ce cas, les bons déplacements sont

$$S = \operatorname{diag}(x_1, \dots, x_n) \quad , \quad T = \operatorname{diag}(y_1, \dots, y_n).$$

On vérifie $\operatorname{rang}_{S,T}(A) = 1$; voyons cela pour $N = 3$:

$$\begin{aligned} \nabla_{S,T}(\mathcal{C}(x,y)) &= \begin{bmatrix} x_1 & & \\ & x_2 & \\ & & x_3 \end{bmatrix} \cdot \begin{bmatrix} \frac{1}{x_1-y_1} & \frac{1}{x_1-y_2} & \frac{1}{x_1-y_3} \\ \frac{1}{x_2-y_1} & \frac{1}{x_2-y_2} & \frac{1}{x_2-y_3} \\ \frac{1}{x_3-y_1} & \frac{1}{x_3-y_2} & \frac{1}{x_3-y_3} \end{bmatrix} \\ &\quad - \begin{bmatrix} \frac{1}{x_1-y_1} & \frac{1}{x_1-y_2} & \frac{1}{x_1-y_3} \\ \frac{1}{x_2-y_1} & \frac{1}{x_2-y_2} & \frac{1}{x_2-y_3} \\ \frac{1}{x_3-y_1} & \frac{1}{x_3-y_2} & \frac{1}{x_3-y_3} \end{bmatrix} \cdot \begin{bmatrix} y_1 & & \\ & y_2 & \\ & & y_3 \end{bmatrix} \\ &= \begin{bmatrix} \frac{x_1}{x_1-y_1} & \frac{x_1}{x_1-y_2} & \frac{x_1}{x_1-y_3} \\ \frac{x_2}{x_2-y_1} & \frac{x_2}{x_2-y_2} & \frac{x_2}{x_2-y_3} \\ \frac{x_3}{x_3-y_1} & \frac{x_3}{x_3-y_2} & \frac{x_3}{x_3-y_3} \end{bmatrix} - \begin{bmatrix} \frac{y_1}{x_1-y_1} & \frac{y_2}{x_1-y_2} & \frac{y_3}{x_1-y_3} \\ \frac{y_1}{x_2-y_1} & \frac{y_2}{x_2-y_2} & \frac{y_3}{x_2-y_3} \\ \frac{y_1}{x_3-y_1} & \frac{y_2}{x_3-y_2} & \frac{y_3}{x_3-y_3} \end{bmatrix} \\ &= \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}. \end{aligned}$$

On summarise :

S	T	Structure	$\mathrm{rang}_{S,T}(A)$
Z_1	Z_0	Toeplitz	2
Z_1	Z_0^T	Hankel	2
$\mathrm{diag}(x_i^{-1})$	Z_0	Vandermonde	1
$\mathrm{diag}(x_i)$	$\mathrm{diag}(y_i)$	Cauchy	1

Notons que ces quattre structures dépendent de $O(N)$ paramètres et que les opérateurs de déplacement ∇ associées sont simples. Les matrices Toeplitz et Hankel peuvent se recupérer à partir de leurs déplacements $\nabla(A)$, les matrices de Cauchy se récupérent à partir de l'opérateur ∇ lui même ; les matrices de Vandermonde peuvent se récupérer soit de l'opérateur ∇ soit de son image $\nabla(A)$.

Une matrice $A \in \mathbb{K}^{N\times N}$ est *type Toeplitz/Hankel/Vandermonde/Cauchy* si pour les déplacements correspondant à ces structures

$$\mathrm{rang}_{S,T}(A) \ll N.$$

0.2. Générateurs. La méthode de rang de déplacement appliquée à une matrice A consiste en trois étapes :

(1) **Compression :** la matrice A est codée avec son déplacement $\nabla(A)$: le rang de $\nabla(A)$ doit être petit pour que la compression soit efficace.
(2) **Opération :** les opérations sur A (multiplication matrice-vecteur, élimination) peuvent se faire à niveau des déplacements avec des techniques adaptées.
(3) **Decompression :** les résultats des opérations se recupérent à partir de leurs déplacements.

LEMME 0.2. *Soit $C \in \mathbb{K}^{M\times N}$ une matrice de rang α, alors il existent $G, B \in \mathbb{K}^{M\times\alpha}$ telles que*

$$C = G \cdot B^T.$$

DÉMONSTRATION. Deux démonstrations possibles :

(1) Considérons la DVS

$$C = V \cdot \Sigma \cdot U^*$$

avec $V \in U(M)$, $U \in U(N)$ et $\Sigma \in \mathbb{K}^{M\times N}$ "diagonal" dont les coefficients sont les valeurs singulières $\sigma_1 \geq \cdots \geq \sigma_{\min(M,N)} \geq 0$. Du fait que $\mathrm{rang}(C) = \alpha$ on a $\sigma_j = 0$ pour $j > \alpha$. Soient

$$v_1, \ldots, v_\alpha \in \mathbb{K}^M \quad , \quad u_1, \ldots, u_\alpha \in \mathbb{K}^N$$

les premières α colonnes de V et de U respectivement, alors

$$C = [v_1 \cdots v_\alpha] \cdot \mathrm{diag}(\sigma_1, \ldots, \sigma_\alpha) \cdot [u_1 \cdots u_\alpha]^*$$

en on peut prendre

(1)
$$G := [v_1 \cdots v_\alpha] \cdot \mathrm{diag}(\sigma_1^{1/2}, \ldots, \sigma_\alpha^{1/2}) \quad , \quad B := \cdot[\overline{u}_1 \cdots \overline{u}_\alpha] \cdot \mathrm{diag}(\sigma_1^{1/2}, \ldots, \sigma_\alpha^{1/2}).$$

(2) Soient $c_1, \ldots, c_N \in \mathbb{K}^N$ les colonnes de C et prenons $g_1, \ldots, g_\alpha \in \mathbb{K}^M$ une base quelconque de l'espace linéaire engendré par ces vecteurs. Soient alors $b_{k,j}$ $(1 \leq k \leq \alpha, 1 \leq j \leq N)$ tels que

(2)
$$c_j = \sum_{k=1}^{\alpha} b_{k,j} g_k \quad \text{pour } 1 \leq j \leq N.$$

On pose alors

$$G := [g_1 \cdots g_\alpha] \quad , \quad B := [b_{j,k}]_{1\leq j\leq N, 1\leq k\leq\alpha};$$

l'équation (2) équivaut à ce que $C = G \cdot B^T$.

□

Une couple (G, B) comme dans le lemme ci-dessus est un système de *générateurs* de la matrice C. Le nombre de coefficients de C est MN et celui des générateurs (G, B) est $\alpha(M+N)$. Donc si α est petit devant M, N, la représentation $G \cdot B$ est nettement plus compacte. La morale à retenir est :

Une matrice à petit rang est computationnallement petite

Les générateurs d'une matrice ne sont nullement uniquement déterminés, et une choix possible est de prendre les $g_1, \ldots, g_\alpha$ parmi les colonnes de C comme dans la deuxième démonstration du lemme ci-dessus. Cela fixe d'avance α colonnes de B (c'est une sous-matrice $\mathbf{1}_\alpha$) et donc le nombre de coefficients non triviaux dans B est $\alpha(N-\alpha)$. Avec cette choix, les générateurs (G, B) seront représentés par $\alpha(M+N-\alpha)$ coefficients ; observons que

$$MN \geq \alpha(M+N-\alpha).$$

Cependant, les générateurs à colonnes orthogonales qu'on obtient *via* la DSV (display 1) sont préferables si l'on veut assurer la stabilité numérique de la représentation de C, *voir* les références dans [**14**, §4.6.1].

En conséquent, les matrices de petit rang seront codées et traitées *via* des générateurs et non pas comme des matrices denses. En particulier, une matrice A à petit rang de déplacement sera codée par des générateurs (G, B) pour le déplacement $\nabla(A)$.

Générateurs pour les structures classiques : on fait toujours le cas $N = 3$, pour N quelconque les formules se généralisant de façon évidente.

(1) Matrices de Toeplitz :

$$\nabla_{Z_1,Z_0}(\mathcal{T}) = \begin{bmatrix} a_1 & & \\ a_2 & & \\ a_0 & a_{-1}-a_2 & a_{-2}-a_1 \end{bmatrix} = \begin{bmatrix} a_1 & \\ a_2 & \\ a_0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & & \\ 0 & a_{-1}-a_2 & a_{-2}-a_1 \end{bmatrix}.$$

(2) Matrices de Hankel :

$$\nabla_{Z_1,Z_0^T}(\mathcal{H}) = \begin{bmatrix} & & a_3 \\ & & a_4 \\ a_0-a_3 & a_1-a_4 & a_2 \end{bmatrix} = \begin{bmatrix} & a_3 \\ & a_4 \\ 1 & a_2 \end{bmatrix} \cdot \begin{bmatrix} a_0-a_3 & a_1-a_4 & \\ & & 1 \end{bmatrix}.$$

(3) Matrices de Vandermonde :

$$\nabla_{S,T}(\mathcal{V}) = \begin{bmatrix} x_1^{-1} & 0 & 0 \\ x_2^{-1} & 0 & 0 \\ x_3^{-1} & 0 & 0 \end{bmatrix} = \begin{bmatrix} x_1^{-1} \\ x_2^{-1} \\ x_3^{-1} \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}.$$

(4) Matrices de Cauchy :

$$\nabla_{S,T}(\mathcal{C}) = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}.$$

0.3. Opérations de base. Passons revue aux propriétés de base, qu'on démontrera en exercice :

LEMME 0.3. *Soient $A, B, S, T, U \in \mathbb{K}^{N\times N}$, alors*

(1) $\nabla_{S,T}(A+B) = \nabla_{S,T}(A) + \nabla_{S,T}(B)$;

(2) $\nabla_{T^T,S^T}(A^T) = \nabla_{S,T}(A)^T$;

(3) $\nabla_{T,S}(A^{-1}) = -A^{-1} \cdot \nabla_{S,T}(A) \cdot A^{-1}$;

(4) $\nabla_{S,U}(A \cdot B) = \nabla_{S,T}(A) \cdot B + A \cdot \nabla_{T,U}(B)$.

En particulier

(1) $\mathrm{rang}_{S,T}(A+B) \leq \mathrm{rang}_{S,T}(A) + \mathrm{rang}_{S,T}(B)$;

(2) $\mathrm{rang}_{T^T,S^T}(A^T) = \mathrm{rang}_{S,T}(A)$;

(3) $\mathrm{rang}_{T,S}(A^{-1}) = \mathrm{rang}_{S,T}(A)$;

(4) $\mathrm{rang}_{S,U}(A \cdot B) \leq \mathrm{rang}_{S,T}(A) + \mathrm{rang}_{T,U}(B)$.

Ces propriétés se généralisent à des matrices rectagulaires des que les dimensions sont compatibles.

Ces formules sont particulièrement sympathiques lorsque $S = T$, par exemple pour les matrices de Toeplitz, où l'on peut prendre $S = T = Z_0$. Dans ce cas, les matrices formées à partir d'autres au moyen d'opérations arithmétiques sont de rang de déplacement controlé. Par exemple, si T_1, T_2, T_3 sont Toeplitz, alors

$$T_1^{-1} T_2 - T_3$$

est de (Z_0, Z_0)-rang au plus 6.

Ces formules nous permettent de calculer de générateurs pour les matrices produites, en termes de générateurs des matrices donnés (sauf pour l'inversion, bien evidémment !). Soient

$$\nabla_{S,T}(A) = G \cdot C^T \quad , \quad \nabla_{S,T}(B) = G' \cdot (C')^T$$

des générateurs de longueur α et β respectivement, alors

$$\nabla_{S,T}(A+B) = \nabla_{S,T}(A) + \nabla_{S,T}(B) = [G|G'] \cdot [C|C']^T$$

où $[G|G'], [C|C'] \in \mathbb{K}^{N \times (\alpha+\beta)}$ est la *concatenation* de matrices G, G' et C, C' respectivement. Similairement pour

$$\nabla_{S,U}(A) = G \cdot C^T \quad , \quad \nabla_{T,U}(B) = G' \cdot (C')^T,$$

on a

$$\nabla_{S,U}(A \cdot B) = \nabla_{S,T}(A) \cdot B + A \cdot \nabla_{T,U}(B) = [G|AG'] \cdot [B^T C|C']^T.$$

Les générateurs ainsi produits ne sont pas forcémment de longueur minimale. Dans la sous-section 2.1 on donne des techniques pour rendre minimale la longueur d'un système de générateurs donné.

Considérons maintenant la décomposition en blocs

$$A = \begin{bmatrix} A_{1,1} & A_{1,2} \\ A_{2,1} & A_{2,2} \end{bmatrix} \begin{matrix} i \\ N-i \end{matrix}$$
$$\begin{matrix} i & N-i \end{matrix}$$

et similairement pour les opérateurs S et T. Le résultat suivant fournit des formules pour le déplacement de chacun des blocs ; la vérification est directe :

LEMME 0.4. *Pour* $1 \leq i, j \leq 2$

$$\nabla_{S_{i,i},T_{j,j}}(A_{i,j}) = \nabla_{S,T}(A)_{i,j} - A_{i,3-j} \cdot T_{3-j,j} + S_{i,3-i} \cdot A_{3-i,j}.$$

Soit

$$\beta_{S,T} := \max\{\mathrm{rang}(S_{1,2}), \mathrm{rang}(S_{2,1}), \mathrm{rang}(T_{1,2}), \mathrm{rang}(T_{2,1})\}$$

le maximum de la dimension des blocs de S et T dehors la diagonale, alors

$$\begin{aligned} \mathrm{rang}_{S_{i,i},T_{j,j}}(A_{i,j}) &\leq \mathrm{rang}(\nabla_{S,T}(A)_{i,j}) + \mathrm{rang}(T_{3-j,j}) + \mathrm{rang}(S_{i,3-i}) \\ &\leq \mathrm{rang}(\nabla_{S,T}(A)_{i,j}) + 2\beta_{S,T}. \end{aligned}$$

Les déplacements S et T sont *quasi-diagonale par blocs* si pour tout $1 \leq i \leq N-1$ les blocs

$$S_{1,2} \quad , \quad S_{2,1} \quad , \quad T_{1,2} \quad , \quad T_{2,1}$$

sont de rang petit devant N. Ce le cas des quattre types de structures classiques : on a alors $\beta_{S,T} \leq 1$. Pour des déplacements quasi-diagonales par blocs, le rang de déplacement des matrices produites par les opérations de somme, multiplication, inversion, transposition, et projection aux blocs, reste controlé.

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

$$A_{(i)} := A_{2,2} - A_{2,1}A_{1,1}^{-1}A_{1,2} \in \mathbb{K}^{(N-i)\times(N-i)}$$

le i-ème complément de Schur de A. On peut obtenir une expression pour le déplacement de $A_{(i)}$ en combinant les lemmes 0.3 et 0.4 ; la proposition suivante nous en fournit une expression plus compacte :

PROPOSITION 0.5. *Soient $A, S, T \in \mathbb{K}^{N\times N}$ et soit*

$$\begin{aligned}\nabla_{S,T}(A) &= \begin{bmatrix} S_{1,1} & S_{1,2} \\ S_{2,1} & S_{2,2} \end{bmatrix} \cdot \begin{bmatrix} A_{1,1} & A_{1,2} \\ A_{2,1} & A_{2,2} \end{bmatrix} - \begin{bmatrix} A_{1,1} & A_{1,2} \\ A_{2,1} & A_{2,2} \end{bmatrix} \cdot \begin{bmatrix} T_{1,1} & T_{1,2} \\ T_{2,1} & T_{2,2} \end{bmatrix} \\ &= \begin{bmatrix} G_1 \\ G_2 \end{bmatrix} \cdot \begin{bmatrix} B_1^T & B_2^T \end{bmatrix},\end{aligned}$$

la décomposition du déplacement et des générateurs dans des blocs de taille i et $N-i$. Supposons $A_{1,1}$ inversible et soit $A_{(i)}$ le complément de Schur correspondant, alors

$$\nabla_{S_{2,2},T_{2,2}}(A_{(i)}) = H \cdot C^T + A_{2,1}A_{1,1}^{-1}S_{1,2}A_{(i)} - A_{(i)}T_{2,1}A_{1,1}^{-1}A_{1,2}$$

avec

$$H = G_2 - A_{2,1} \cdot A_{1,1}^{-1} \cdot G_1 \quad , \quad C = B_2 - (A_{1,1}^{-1} \cdot A_{1,2})^T \cdot B_1 \quad \in \mathbb{K}^{(N-i)\times\alpha}. \tag{3}$$

DÉMONSTRATION. On a

$$A = \begin{bmatrix} A_{1,1} & A_{1,2} \\ A_{2,1} & A_{2,2} \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{1}_i & \\ A_{2,1}A_{1,1}^{-1} & \mathbf{1}_{N-i} \end{bmatrix}}_{V} \cdot \begin{bmatrix} A_{1,1} & \\ & A_{(i)} \end{bmatrix} \cdot \underbrace{\begin{bmatrix} \mathbf{1}_i & A_{1,1}^{-1}A_{1,2} \\ & \mathbf{1}_{N-i} \end{bmatrix}}_{W}.$$

Les matrices V et W sont inversibles et on a

$$\begin{aligned}V^{-1}SV &= \begin{bmatrix} \mathbf{1}_i & \\ -A_{2,1}A_{1,1}^{-1} & \mathbf{1}_{N-i} \end{bmatrix} \cdot \begin{bmatrix} S_{1,1} & S_{1,2} \\ S_{2,1} & S_{2,2} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{1}_i & \\ A_{2,1}A_{1,1}^{-1} & \mathbf{1}_{N-i} \end{bmatrix} \\ &= \begin{bmatrix} * & * \\ * & S_{2,2} - A_{2,1}A_{1,1}^{-1}S_{1,2} \end{bmatrix}\end{aligned}$$

et similairement

$$WTW^{-1} = \begin{bmatrix} * & * \\ * & T_{2,2} - T_{2,1}A_{1,1}^{-1}A_{1,2} \end{bmatrix};$$

donc

$$\begin{aligned}V^{-1}(SA - AT)W^{-1} &= (V^{-1}SV)(V^{-1}AW^{-1}) - (V^{-1}AW^{-1})(WTW^{-1}) \\ &= \begin{bmatrix} * & * \\ * & S_{2,2} - A_{2,1}A_{1,1}^{-1}S_{1,2} \end{bmatrix} \cdot \begin{bmatrix} A_{1,1} & \\ & A_{(i)} \end{bmatrix} \\ &\quad - \begin{bmatrix} A_{1,1} & \\ & A_{(i)} \end{bmatrix} \cdot \begin{bmatrix} * & * \\ * & T_{2,2} - T_{2,1}A_{1,1}^{-1}A_{1,2} \end{bmatrix} \\ &= \begin{bmatrix} * & * \\ * & \nabla_{S_{2,2}-A_{2,1}A_{1,1}^{-1}S_{1,2},T_{2,2}-T_{2,1}A_{1,1}^{-1}A_{1,2}}(A_{(i)}) \end{bmatrix}.\end{aligned}$$

De l'autre côté

$$V^{-1}(SA-AT)W^{-1} = \begin{bmatrix} \mathbf{1}_i & \\ -A_{2,1}A_{1,1}^{-1} & \mathbf{1}_{N-i} \end{bmatrix} \cdot \begin{bmatrix} G_1 \\ G_2 \end{bmatrix} \cdot \begin{bmatrix} B_1^T & B_2^T \end{bmatrix} \cdot \begin{bmatrix} \mathbf{1}_i & -A_{1,1}^{-1}A_{1,2} \\ & \mathbf{1}_{N-i} \end{bmatrix}$$
$$= \begin{bmatrix} G_1 \\ H \end{bmatrix} \cdot \begin{bmatrix} B_1^T & C^T \end{bmatrix}$$

avec H, C comme dans les formules (3) ; en identifiant les blocs $(2,2)$ on obtient

$$\nabla_{S_{2,2}-A_{2,1}A_{1,1}^{-1}S_{1,2},T_{2,2}-T_{2,1}A_{1,1}^{-1}A_{1,2}}(A_{(i)})$$
$$= \nabla_{S_{2,2},T_{2,2}}(A_{(i)}) - A_{2,1}A_{1,1}^{-1}S_{1,2}A_{(i)} + A_{(i)}T_{2,1}A_{1,1}^{-1}A_{1,2} = H \cdot C^T.$$

□

Les déplacements S et T sont respectivement *quasi-triangulaires par blocs* inférieure et supérieure si les blocs

$$S_{1,2} \quad , \quad T_{2,1}$$

sont de rang petit devant N. Soit

$$\gamma_{S,T} := \max\{\text{rang}(S_{1,2}), \text{rang}(T_{2,1})\},$$

alors

$$\text{rang}_{S_{2,2},T_{2,2}}(A_{(i)}) \leq \text{rang}_{S,T}(A) + \text{rang}(S_{1,2}) + \text{rang}(T_{2,1}) \leq \text{rang}_{S,T}(A) + 2\gamma_{S,T};$$

c'est-à-dire le rang déplacement du complément de Schur reste controlé dès que les déplacements sont quasi-triangulaires par blocs. Voici le comportement des quattre structures classiques

Structure	$\beta_{S,T}$	$\gamma_{S,T}$
Toeplitz	1	1
Hankel	1	1
Vandermonde	1	0
Cauchy	0	0

Pour le cas des opérateurs S et T respectivement triangulaire inférieur et supérieur on obtient le résultat le plus net :

Corollaire 0.6. *Avec les notations de la proposition 0.6, suppossons S, T respectivement triangulaire inférieur et supérieur par blocs, alors*

$$\nabla_{S_{2,2},T_{2,2}}(S_i) = H \cdot C^T$$

avec

$$(4) \quad H = G_2 - A_{2,1} \cdot A_{1,1}^{-1} \cdot G_1 \quad , \quad C = B_2 - (A_{1,1}^{-1} \cdot A_{1,2})^T \cdot B_1 \quad \in \mathbb{K}^{(N-i)\times\alpha}.$$

En particulier

$$\text{rang}_{S_{2,2},T_{2,2}}(S_i) \leq \text{rang}_{S,T}(A).$$

Ainsi, dans cette situation le rang de déplacement n'augmente pas par l'opération de prendre complément de Schur, et de plus grâce aux formules (4) on peut faire le passage

$$(G, B) \to (H, C)$$

sans avoir à expliciter le complément de Schur $A_{(i)}$ lui même. Ce genre de propriété est la clé de tous les algorithmes dans cette domaine. Notons en passant que ces formules d'actualisation ne font intervernir les matrices de déplacement.

1. Reconstruction

Dans cette section on étude l'inversion de l'opérateur de déplacement, necessaire pour l'étape de decompression dans les algorithmes de résolution pour les matrices structurées. Le problème est donc celui de résoudre en A l'équation

$$S \cdot A - A \cdot T = \nabla \tag{5}$$

pour $\nabla \in \mathbb{K}^{M\times N}$ donnée. Cette équation linéaire a un nom, c'est l'*équation de Sylvester continue*, et sa solvabilité admets une caractérisation sympathique (proposition 1.2 ci-dessous).

D'abord on linéarise le problème ; pour cela on a besoin d'un peu de notation. Pour une matrice $A \in \mathbb{K}^{M\times N}$ soient $a_1, \ldots, a_N \in \mathbb{K}^M$ ses colonnes et posons

$$\mathrm{Vect}(A) = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_N \end{bmatrix} \in \mathbb{K}^{MN}$$

le *vecteur colonne* fait des colonnes de A empilées les unes sur les autres. Pour $C \in \mathbb{K}^{M\times N}$ et $D \in \mathbb{K}^{P\times Q}$, le *produit tensoriel* $C \otimes D$ est

$$C \otimes D = \begin{bmatrix} c_{1,1}D & \cdots & c_{1,N}D \\ \vdots & & \vdots \\ c_{M,1}D & \cdots & c_{M,N}D \end{bmatrix} \in \mathbb{K}^{MP\times NQ}.$$

Proposition 1.1. *L'équation $S \cdot A - A \cdot T = \nabla$ équivaut à*

$$(\mathbf{1}_N \otimes S + T^T \otimes \mathbf{1}_N) \cdot \mathrm{Vect}(A) = \mathrm{Vect}(\nabla). \tag{6}$$

Démonstration. Le processus de vectorisation est une bijection linéaire, donc l'équation de Sylvester continue équivaut à

$$\mathrm{Vect}(S \cdot A) - \mathrm{Vect}(A \cdot T) = \mathrm{Vect}(\nabla).$$

Soient $a_1, \ldots, a_N \in \mathbb{K}^N$ les colonnes de A, alors

$$(\mathbf{1}_N \otimes S)\mathrm{Vect}(A) = \begin{bmatrix} S & & \\ & \ddots & \\ & & S \end{bmatrix} \begin{bmatrix} a_1 \\ \vdots \\ a_N \end{bmatrix} = \mathrm{Vect}(S \cdot A), \tag{7}$$

et pour l'autre terme,

$$(T^T \otimes \mathbf{1}_M)\mathrm{Vect}(A) = \begin{bmatrix} t_{1,1}\mathbf{1}_M & \cdots & t_{N,1}\mathbf{1}_M \\ \vdots & & \vdots \\ t_{1,N}\mathbf{1}_M & \cdots & t_{N,N}\mathbf{1}_M \end{bmatrix} \begin{bmatrix} a_1 \\ \vdots \\ a_N \end{bmatrix} = \mathrm{Vect}(A \cdot T), \tag{8}$$

d'où on conclut. □

Proposition 1.2. *Les valeurs propres de $\nabla_{S,T}$ sont les nombres de la forme $\sigma - \tau$, où σ et τ sont des valeurs propres de S et de T, respectivement.*
En particulier, $\nabla_{S,T}$ est inversible si et seulement si $\mathrm{Spec}(S) \cap \mathrm{Spec}(T) = \emptyset$.

Démonstration. Quitte à passer à $\mathbb{C}$, les matrices S et T sont conjuguées de matrices triangulaires supérieure et inférieure, respectivement. Écrivons des telles décompositions

$$S = E \cdot U \cdot E^{-1}, \quad T = F \cdot L \cdot F^{-1}$$

avec $E, U \in \mathbb{C}^{M\times M}$ et $F, L \in \mathbb{C}^{N\times N}$, E, F inversibles et U, L triangulaires supérieure et inférieure, respectivement. On a

$$E^{-1}\cdot\nabla_{S,T}(A)\cdot F^{-1} = (E^{-1}SE)(E^{-1}AF^{-1}) - (E^{-1}AF^{-1})(FTF^{-1}) \tag{9}$$
$$= U(E^{-1}AF^{-1}) - (E^{-1}AF^{-1})L = \nabla_{U,L}(E^{-1}AF^{-1})$$

donc les valeurs propres de $\nabla_{S,T}$ coïncident avec ceux de $\nabla_{U,L}$.

Les expressions (7) et (8) appliquées à U, L à la place de S, T, montrent que la matrice $\mathbf{1}_n \otimes U + L^T \otimes \mathbf{1}_m$ est triangulaire supérieure et donc ses valeurs propres sont les éléments dans la diagonale principale. Cette diagonale est

$$(\mathrm{diag}(U), \dots, \mathrm{diag}(U)) - (\ell_{1,1}, \dots, \ell_{1,1}, \ell_{2,2}, \dots, \ell_{2,2}, \dots, \ell_{n,n}, \dots, \ell_{n,n}) \in \mathbb{C}^{MN},$$

et ses éléments sont de la forme $\sigma - \tau$ avec $\sigma \in \mathrm{Spec}(U) = \mathrm{Spec}(S)$ et $\tau \in \mathrm{Spec}(L) = \mathrm{Spec}(T)$. □

On isole l'identité 9 dans la preuve ci-dessus, pour utilisation ultérieure :

LEMME 1.3. *Soient $S = E\cdot U\cdot E^{-1}$ et $T = F\cdot L\cdot F^{-1}$ avec $E, U \in \mathbb{C}^{M\times M}$ et $F, L \in \mathbb{C}^{N\times N}$, alors*

$$\nabla_{S,T}(A) = E\cdot\nabla_{U,L}(E^{-1}AF^{-1})\cdot F.$$

Reconstruction de matrices type Cauchy :

PROPOSITION 1.4. *Soient $x, y \in \mathbb{K}^N$ tels que $x_i \neq y_j$ pour tout $1 \leq i, j \leq N$, et $A \in \mathbb{K}^{N\times N}$ tel que*

$$\mathrm{diag}(x)\cdot A - A\cdot\mathrm{diag}(y) = G\cdot B^T$$

avec $G, B \in \mathbb{K}^{N\times\alpha}$, alors

$$A = \sum_{k=1}^{\alpha}\mathrm{diag}(g_k)\cdot\mathcal{C}(x,y)\cdot\mathrm{diag}(b_k).$$

DÉMONSTRATION. Soit C la matrice définie para l'expression de droite, alors

$$\nabla(C) = \sum_{k=1}^{\alpha}\mathrm{diag}(g_k)\cdot\nabla(\mathcal{C}(x,y))\cdot\mathrm{diag}(b_k)$$

puisque les déplacements sont diagonaux et donc commutent avec $\mathrm{diag}(g_k)$ et $\mathrm{diag}(b_k)$. Ainsi

$$\nabla(C) = \sum_{k=1}^{\alpha}\mathrm{diag}(g_k)\cdot\begin{bmatrix}1\\ \vdots\\ 1\end{bmatrix}\cdot\begin{bmatrix}1\cdots 1\end{bmatrix}\cdot\mathrm{diag}(b_k) = \sum_{k=1}^{\alpha} g_k\cdot b_k^T = G\cdot B^T = \nabla(A),$$

ce qui entraîne $A = C$ car $\mathrm{Spec}(\mathrm{diag}(x))$ et $\mathrm{Spec}(\mathrm{diag}(y))$ sont disjoints. □

En particulier, si S, T sont diagonalizables à spectre disjoint, on peut résoudre $SA - AT = \nabla$ par reduction au cas des type Cauchy.

Reconstruction des matrices type Vandermonde :

PROPOSITION 1.5. *Soient $x_1, \dots, x_N \in \mathbb{K}^\times$ et $A \in \mathbb{K}^{N\times N}$ such that*

$$\mathrm{diag}(x_i^{-1})_{1\leq i\leq N}\cdot A - A\cdot Z_0 = G\cdot B^T$$

avec $G, B \in \mathbb{K}^{N\times\alpha}$*, alors*

$$A = \sum_{k=1}^{\alpha} \mathrm{diag}(x_i g_{i,k})_{1\le i\le N} \cdot \mathcal{V}(x) \cdot \begin{bmatrix} b_{1,k} & b_{2,k} & \cdots & b_{N,k} \\ & b_{1,k} & \cdots & b_{N-1,k} \\ & & \ddots & \vdots \\ & & & b_{1,k} \end{bmatrix}.$$

Démonstration. On a

$$\mathcal{B}_k := \begin{bmatrix} b_{1,k} & b_{2,k} & \cdots & b_{N,k} \\ & b_{1,k} & \cdots & b_{N-1,k} \\ & & \ddots & \vdots \\ & & & b_{1,k} \end{bmatrix} = b_{1,k}\mathbf{1}_N + b_{2,k}Z_0 + \cdots + b_{N,k}Z_0^{N-1}$$

donc $\mathcal{B}_k$ commute avec Z_0. Soit C la matrice définie para l'expression de droite, alors

$$\begin{aligned}
\nabla(C) &= \sum_{k=1}^{\alpha} \big(\mathrm{diag}(x_i g_{i,k})_i \cdot \mathrm{diag}(x_i^{-1})_i \cdot \mathcal{V}(x) \cdot \mathcal{B}_k - \mathrm{diag}(x_i g_{i,k})_i \cdot \mathcal{V}(x) \cdot Z_0 \cdot \mathcal{B}_k\big) \\
&= \sum_{k=1}^{\alpha} \mathrm{diag}(x_i g_{i,k})_i \cdot \begin{bmatrix} x_1^{-1} \\ x_2^{-1} \\ \vdots \\ x_N^{-1} \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix} \cdot \mathcal{B}_k \\
&= \sum_{k=1}^{\alpha} g_k \cdot b_k^T \\
&= G \cdot B^T = \nabla(A),
\end{aligned}$$

ce qui entraîne $A = C$, car

$$\mathrm{Spec}(\mathrm{diag}(x_i^{-1})_i) = \{x_1^{-1}, \dots, x_N^{-1}\} \quad , \quad \mathrm{Spec}(Z_0) = \{0\}$$

sont disjoints. □

Le circulant est diagonalisable *via* la TFD :

Proposition 1.6. *Soit* $\omega = e^{-2i\pi/N}$ *et* $F_N = [\omega^{ij}]_{0\le i,j\le N-1}$ *la matrice de la TFD, alors*

$$Z_1 = F_N \cdot \mathrm{diag}(\omega^j)_{0\le j\le N-1} \cdot F_N^{-1}.$$

Démonstration. On a

$$Z_1 v = \lambda v \quad \Longleftrightarrow \quad \begin{aligned} v_2 &= \lambda v_1, \\ v_3 &= \lambda v_2, \\ &\vdots \\ v_1 &= \lambda v_N; \end{aligned}$$

d'où

$$\lambda = \omega^j \quad , \quad v = (1, \omega^j, \dots, (\omega^j)^{N-1})$$

pour quelque $0 \le j \le N-1$, donc

$$Z_1 \cdot F_N = F_N \cdot \mathrm{diag}(1, \omega, \cdots, \omega^{N-1}).$$

□

Ceci entraîne la réduction des matrices type Toeplitz au type Vandermonde : posons $D := \mathrm{diag}(\omega^j)_{0\le j\le N-1}$, pour $A \in \mathbb{K}^{N\times N}$

$$\nabla_{Z_1,Z_0}(A) = F_N \cdot \nabla_{D,Z_0}(F_N^{-1} \cdot A) = \frac{1}{N} F_N \cdot \nabla_{D,Z_0}(F_N^{*} \cdot A),$$

donc si A est type Toeplitz alors $F_N^* \cdot A$ est type Vandermonde avec le même rang de déplacement, et si $\nabla_{Z_1,Z_0}(A) = G \cdot B^T$ alors

$$\nabla_{D,Z_0}(F_N^* \cdot A) = (NF_N \cdot G) \cdot B^T$$

et de plus la multiplication $F_N \cdot G$ se fait en $1.5\alpha N \log(N)$ ops avec la TFR.

Posons

$$J = \begin{bmatrix} & & 1 \\ & \diagup & \\ 1 & & \end{bmatrix}.$$

Ceci est une permutation et on a

$$JJ^T = J^2 = 1 \quad , \quad Z_0^T = J \cdot Z_0 \cdot J^{-1}.$$

Avec ceci on réduit facilement les matrices type Hankel au type Toeplitz, et *a fortiori* au type Vandermonde : pour $A \in \mathbb{K}^{N\times N}$,

$$\nabla_{Z_1,Z_0^T}(A) = \nabla_{Z_1,Z_0}(A \cdot J) \cdot J$$

donc si $\nabla_{Z_1,Z_0^T}(A) = G \cdot B^T$ alors

$$\nabla_{Z_1,Z_0}(A \cdot J) = G \cdot (B^T \cdot J).$$

2. Algorithmes rapides

Les outils sont en place pour la version "rapide" de l'algorithme de résolution de matrices structurées. L'idée consiste en opérer à niveau des générateurs et non pas sur les matrices elles mêmes. Seulement on se permets de reconstruire et d'opérer avec des petits morceaux des matrices sous-jacentes.

Pour la décomposition LU sans pivotage d'une matrice A fortément inversible, l'algorithme consiste à calculer les *générateurs* des compléments de Schur $A(i)$ successifs, puis de reconstruire à chaque étape les premières ligne et colonne de $A(i)$ a fin de produire les facteurs L et U. En accord avec ça, on suppose la matrice d'entrée A codée par ses générateurs et non pas dans la forme dense. Dans une première approche on se restreindra à des déplacements $S, T \in \mathbb{K}^{N\times N}$ triangulaires inférieure et supérieure respectivement (pour pouvoir appliquer la proposition 0.6) et tels que $\mathrm{Spec}(S) \cap \mathrm{Spec}(T) = \emptyset$, pour que la reconstruction soit possible.

Algorithme de décomposition LU des matrices structurées :

Entrée : $S, T \in \mathbb{K}^{N\times N}$ triangulaires inférieure et supérieure respectivement telles que $\mathrm{Spec}(S) \cap \mathrm{Spec}(T) = \emptyset$;

$G(0), B(0) \in \mathbb{K}^{N\times\alpha}$ générateurs d'une matrice $A = A(0)$ fortément inversible ;

Sortie : $L, U \in \mathbb{K}^{N\times N}$ décomposition LU de A.

For i from 1 to $N-1$ do

(1) reconstruire le pivot et les premières ligne et colonne

$$A(i)_{1,1} \leftarrow a_{1,1} \quad , \quad A(i)_{1,2} \leftarrow \begin{bmatrix} a_{i,i+1} & \cdots & a_{i,N} \end{bmatrix} \quad , \quad A(i)_{2,1} \leftarrow \begin{bmatrix} a_{i+1,i} \\ \vdots \\ a_{N,i} \end{bmatrix}$$

de $A(i) \in \mathbb{K}^{(N-i+1)\times(N-i+1)}$ vérifiant $\nabla_i(A(i)) = G(i) \cdot B(i)^T$;

(2) construire la i-ème colonne de L et la i-ème ligne de U :

$$L_{i,i} \leftarrow 1 \quad , \quad L_{i+1\le j\le N,i} \leftarrow A(i)_{1,1}^{-1} A(i)_{2,1};$$
$$U_{i,i} \leftarrow A(i)_{1,1} \quad , \quad U_{i,i+1\le j\le N} \leftarrow A(i)_{1,2};$$

(3) actualiser les générateurs :

$$G(i+1) \leftarrow G(i)_2 - A(i)_{2,1} \cdot A(i)_{1,1}^{-1} \cdot G(i)_1,$$
$$B(i+1)^T \leftarrow B(i)_2^T - B(i)_1^T \cdot A(i)_{1,1}^{-1} \cdot A(i)_{1,2};$$

od ;

$L_{N,N} \leftarrow 1, \quad U_{N,N} \leftarrow G(N) \cdot B(N)$ **;**

end.

Pour introduire le pivotage, considerons l'effet d'une permutation P sur le déplacement :

$$P \cdot \nabla_{S,T}(A) = P \cdot S \cdot A - P \cdot A \cdot T = P \cdot S \cdot P^* \cdot (P \cdot A) - (P \cdot A) \cdot T = \nabla_{PSP^*,T}(P \cdot A). \tag{10}$$

Donc la matrice après pivotage $P \cdot A$ reste structurée, seulement S doit être conjugée par la permutation. Pour que $P \cdot S \cdot P^*$ reste triangulaire inférieure pour toute permutation P, il faut que S soit diagonale. Le pivotage partiel restreint encore l'application de l'algorithme rapide de décomposition LU à des deplacements S *diagonal* et T *triangulaire supérieure*. Le pivotage partiel introduit le pas supplémentaire

(0.5) reconstruire la première colonne

$$c_i \leftarrow \begin{bmatrix} a_{i,i} \\ \vdots \\ a_{i,N} \end{bmatrix}$$

de $A(i) \in \mathbb{K}^{(N-i+1)\times(N-i+1)}$ **vérifiant** $\nabla_i(A(i)) = G(i) \cdot B(i)^T$ **; déterminer** $i \le k \le N$ **tel que** $|c_{k,i}|$ **soit maximal.**

Garder P_i **la permutation correspondante ; interchanger les lignes** i **et** k **dans le générateur** $G(i)$ **et conjuger** S **suivant** P_i**.**

Si l'on fait du pivotage, le pas (1) se réduit au calcul de la première ligne de $A(i)$.

Il nous reste encore d'expliciter comment fait-on le pas de reconstruction des premières ligne et colonne de $A(i)$. Voici comment on fait pour le type Cauchy :

PROPOSITION 2.1. *Soient $x, y \in \mathbb{K}^N$ tels que $x_i \neq y_j$ pour tout $1 \le i, j \le N$, et $A \in \mathbb{K}^{N\times N}$ tel que*

$$\operatorname{diag}(x) \cdot A - A \cdot \operatorname{diag}(y) = G \cdot B^T$$

avec $G, B \in \mathbb{K}^{N\times\alpha}$, alors

$$A_{i,j} = \frac{1}{x_i - y_j} \sum_{k=1}^{\alpha} g_{i,k} b_{j,k}.$$

DÉMONSTRATION. C'est une vérification directe à partir de la proposition 1.5 ; faisons-le pour $N = 3$. Soit $1 \le k \le \alpha$, alors

$$\begin{bmatrix} g_{1,k} & & \\ & g_{2,k} & \\ & & g_{3,k} \end{bmatrix} \cdot \begin{bmatrix} \frac{1}{x_1-y_1} & \frac{1}{x_1-y_2} & \frac{1}{x_1-y_3} \\ \frac{1}{x_2-y_1} & \frac{1}{x_2-y_2} & \frac{1}{x_2-y_3} \\ \frac{1}{x_3-y_1} & \frac{1}{x_3-y_2} & \frac{1}{x_3-y_3} \end{bmatrix} \cdot \begin{bmatrix} b_{1,k} & & \\ & b_{2,k} & \\ & & b_{3,k} \end{bmatrix}$$
$$= \begin{bmatrix} \frac{g_{1,k}b_{1,k}}{x_1-y_1} & \frac{g_{1,k}b_{2,k}}{x_1-y_2} & \frac{g_{1,k}b_{3,k}}{x_1-y_3} \\ \frac{g_{2,k}b_{1,k}}{x_2-y_1} & \frac{g_{2,k}b_{2,k}}{x_2-y_2} & \frac{g_{2,k}b_{3,k}}{x_2-y_3} \\ \frac{g_{3,k}b_{1,k}}{x_3-y_1} & \frac{g_{3,k}b_{2,k}}{x_3-y_2} & \frac{g_{3,k}b_{3,k}}{x_3-y_3} \end{bmatrix}$$

donc pour $1 \leq i, j \leq 3$

$$A_{i,j} = \frac{1}{x_i - y_j} \sum_{k=1}^{\alpha} g_{i,k} b_{j,k}.$$

□

Voici le type Vandermonde :

LEMME 2.2. *Soient $x_1, \ldots, x_N \in \mathbb{K}^\times$ et $A \in \mathbb{K}^{N \times N}$ tels que*

$$\mathrm{diag}(x_i^{-1})_{1 \leq i \leq N} \cdot A - A \cdot Z_0 = G \cdot B^T$$

avec $G, B \in \mathbb{K}^{N \times \alpha}$, alors

$$A_{i,1} = \sum_{k=1}^{\alpha} g_{i,k} x_k b_{1,k} \quad , \quad i = 1, \ldots, N;$$

$$A_{1,j} = x_1 A_{1,j-1} + \sum_{k=1}^{\alpha} g_{1,k} x_1 b_{j,k} \quad , \quad j = 2, \ldots, N.$$

DÉMONSTRATION. C'est une vérification directe à partir de la proposition 1.5 ; faisons-le pour $N = 3$. Soit $1 \leq k \leq \alpha$, alors

$$\begin{bmatrix} g_{1,k}x_1 & & \\ & g_{2,k}x_2 & \\ & & g_{3,k}x_3 \end{bmatrix} \cdot \begin{bmatrix} 1 & x_1 & x_1^2 \\ 1 & x_2 & x_2^2 \\ 1 & x_3 & x_3^2 \end{bmatrix} \cdot \begin{bmatrix} b_{1,k} & b_{2,k} & b_{3,k} \\ & b_{1,k} & b_{2,k} \\ & & b_{1,k} \end{bmatrix}$$

$$= \begin{bmatrix} g_{1,k}x_1b_{1,k} & g_{1,k}x_1b_{2,k} + g_{1,k}x_1^2b_{1,k} & g_{1,k}x_1b_{3,k} + g_{1,k}x_1^2b_{2,k} + g_{1,k}x_1^3b_{1,k} \\ g_{2,k}x_2b_{1,k} & * & * \\ g_{3,k}x_3b_{1,k} & * & * \end{bmatrix};$$

à partir d'ici on vérifie facilement les formules proposées. □

Les conditions demandés aux déplacements restreint l'application de cet algorithme aux types Vandermonde et Cauchy, parmi les quattre structures classiques. Pour sauter cette restriction, on peut réduire le type Hankel au type Toeplitz en post-multiplicant la matrice par une permutation convénable, et réduire le type Toeplitz au type Vandermonde en preconditionnant par une TFD. Dans les sections suivantes, on généralisara l'algorithme à des déplacements S quasi-diagonal par blocs et T quasi-triangulaire par blocs, ce qu'en particulier permets une approche unifiée à les quattre structures classiques.

PROPOSITION 2.3. *Le coût de l'algorithme de décomposition PLU pour une matrice $A \in \mathbb{K}^{N \times N}$ type Toeplitz/Hankel/Vadermonde/Cauchy avec* $\mathrm{rang}_{S,T}(A) \leq \alpha$ *est de*

$$O(\alpha N^2) \quad \text{ops}.$$

DÉMONSTRATION. On fait la démonstration en détail pour la structure type Vandermonde. Pour chaque $1 \leq i \leq N - 1$ on fait

(1) $2\alpha(N - i + 1) + \alpha - 1$ ops pour la reconstruction de la première colonne du complément de Schur $A(i)$;
(2) le pivotage demande $N - i + 1$ comparaisons ;
(3) la reconstruction de la première ligne de $A(i)$ demande $2\alpha(N-i+1)+\alpha+1$ ops ;
(4) le calcul des générateurs du complément de Schur suivant $A(i+1)$ demande $\alpha + 2\alpha(N - i + 1)$.

Le coût total s'estime en

$$\sum_{i=1}^{N-1}(2\alpha(N-i+1)+\alpha-1)+(N-i+1)+(2\alpha(N-i+1)+\alpha+1)$$
$$+(\alpha+2\alpha(N-i+1))=\left(3\alpha+\frac{1}{2}\right)N^2+O(\alpha N).$$

□

Un point important à adresser c'est la stabilité de la méthode proposé, à ce respect *voir* [**13**]. Aussi il serait intéressant d'estimer la complexité *en exacte*.

Comme direction de recherche : étendre ces algorithmes à des matrices structurées autres que les classes classiques. Notamment, pour les matrices Toeplitz bloc Toeplitz ou de Toeplitz bloc Toeplitz bloc Toeplitz, qui apparaissent de façon naturelle dans la discrétisation des EDP elliptiques à coefficients constants sur une grille uniforme.

EXERCICE 5.1. ◁ Étendre l'algorithme de décomposition PLU à des matrices structurées rectangulaires quelconques. ▷

2.1. Compression de générateurs. Typiquement, la longueur des générateurs augmente avec les opérations de somme, multiplications, projections aux blocs et complément de Schur, ce qui conduit à des générateurs trop longs pour des matrices qui peuvent être à petit rang de déplacement. Cette situation apparaîtra lorsqu'on voudra étendre l'algorithme rapide à des déplacements S et T respectivement quasi-diagonal par blocs et quasi-triangulaire par blocs : l'actualisation des générateurs des compléments de Schur successifs ne sera plus directe, et on devra appliquer les formules (3). Ceci produira des générateurs de plus en plus longs au cours du processus d'élimination, or on peut démontrer que tous les compléments de Schur considérés sont à rang de déplacement borné. Ce phenomène sera encore plus marqué dans l'obtention d'algorithmes super-rapides.

Il y a deux approches pour la compression des générateurs, suivant que l'on est dans un contexte numérique ou algébrique. Dans le premier cas, on préférera le calcul des générateurs orthogonaux *via* la DVS comme dans le display (1). Si par contre on est sur un corps $\mathbb{F}$ par forcément égal à $\mathbb{R}$ ou $\mathbb{C}$, on fera la compression *via* l'algorithme d'élimination. Soit $A \in \mathbb{F}^{N\times N}$ une matrice à rang de déplacement α, et $G, B \in \mathbb{F}^{N\times\ell}$ des générateurs de longueur $\ell > \alpha$. Considérons la décomposition PLU des générateurs

$$G^T = P\cdot L\cdot U \quad , \quad B = P'\cdot L'\cdot U'$$

avec $P, P' \in \mathbb{F}^{\ell\times\ell}$ des permutations, $L, L' \in \mathbb{F}^{\ell\times\ell}$ triangulaires inférieure avec 1s dans la diagonal, et $U, U' \in \mathbb{F}^{\ell\times N}$ triangulaires supérieure. On peut lire le rang de déplacement de A dans U et U' :

$$\alpha = \mathrm{rang}_{S,T}(A) = \min\{\mathrm{rang}(U), \mathrm{rang}(U')\},$$

et ce rang coïncide avec le nombre de lignes non nulles dans U et U' respectivement. On vérifie aisément

$$\nabla_{S,T}(A) = [u_1^T \cdots u_\alpha^T]\cdot V\cdot[u_1' \cdots u_\alpha']$$

où u_i (resp. u_i') déssigne la i-ème ligne de U (resp. u_i') et V est le bloc principal de taille $\alpha\times\alpha$ de $L^T\cdot P^T\cdot P'\cdot L' \in \mathbb{F}^{\ell\times\ell}$. Les matrices

$$H := [u_1^T \cdots u_\alpha^T] \quad , \quad C^T := V\cdot[u_1' \cdots u_\alpha']$$

forment un système de générateurs de longueur minimale α. On vérifie que le coût de calcul de ces générateurs est de $O(\ell^2 N)$ ops ; on renvoie à [**14**, §4.6] pour les détails.

2.2. Extension de l'algorithme rapide. Pour longtemps, c'était un fait accepté que la mise en œuvre de la méthode de rang de déplacement était restreinte à S triangulaire inférieure (ou diagonal si besoin il y avait de pivotage) et T triangulaire supérieure.

Comme on l'a déjà vu, pour les déplacements quasi-diagonales le rang des compléments de Schur reste uniformément borné, et on peut actualiser les générateurs de ces compléments de Schur [**14**, § 4.6]. De plus, la quasi-diagonalité est (essentiellement) invariante par conjugaison par des permutations, donc on peut incorporer pivotage partiel ou total pour rendre l'algorithme numériquement stable.

Cependant, pour un souci de simplicité dans l'exposition, on se centrera dans le cas des déplacements triangulaires supérieure (ou carrément diagonal) et inférieure respectivement.

Dans l'article [**7**] on montre comment la méthode s'étends à des déplacement S et T Hessenberg inférieure et supérieure respectivement. On dit que une matrice S est dîte *Hessenberg inférieure* (resp. *Hessenberg supérieure*) si $S_{i,j} = 0$ pour $j \geq i+2$ (resp. si $S_{i,j} = 0$ pour $i \geq j+2$) c'est-à-dire si S est triangulaire sauf pour les diagonales secondaires. Malhereusement l'algorithme de Heinig et Olshevsky n'admets pas de stratégie de pivotage, et par conséquent est numériquement inestable.

L'algorithme de décomposition LU dans cette section admets des versions hermitiennes, au moins pour le cas de structure type Toeplitz, *voir* [**10**].

2.3. Algorithmes super-rapides. Le premier algorithme super-rapide fut proposé par Morf citemorf80 (*voir* aussi [**11**]) et Bitmead et Anderson [**1**] pour la résolution de systèmes type Toeplitz et Hankel, en $O(N \log^2(N))$ ops. Ceci résulte de la combinaison de l'algorithme de type "divide-and-conquer" de Strassen pour l'inversion de matrices, avec l'idée de rang de déplacement.

Dans ce qui suit on considérera le cas d'une matrice fortément inversible. Le cas général se réduit probabilistiquement à ceci soit par symmétrization soit avec l'application d'un preconditionneur structuré [**14**, Ch. 5].

Soit $A \in \mathbb{K}^{N\times N}$ une matrice fortément inversible. Pour $1 \leq i \leq N$ considérons sa décomposition en blocs

$$A = \begin{matrix} \begin{bmatrix} A_{1,1} & A_{1,2} \\ A_{2,1} & A_{2,2} \end{bmatrix} & \begin{matrix} i \\ N-i \end{matrix} \\ \begin{matrix} i & N-i \end{matrix} \end{matrix} ;$$

alors

$$A = \begin{bmatrix} \mathbf{1}_i & \\ A_{2,1}A_{1,1}^{-1} & \mathbf{1}_{N-i} \end{bmatrix} \cdot \begin{bmatrix} A_{1,1} & A_{1,2} \\ & S_i \end{bmatrix} \tag{11}$$

où

$$S_i = A_{2,2} - A_{2,1}A_{1,1}^{-1}A_{1,2}$$

est le i-ème complément de Schur.

LEMME 2.4. *Soit $A \in \mathbb{K}^{N\times N}$ une matrice fortément inversible, alors $A_{1,1}$ et S_i sont fortément inversibles aussi.*

DÉMONSTRATION. Pour $1 \leq k \leq i$ la factorisation 11 entraîne

$$\det(A^{(k)}) = \det(A_{1,1}^{(k)} \neq 0$$

donc $A_{1,1}$ est fortément inversible. Pour $i \leq k \leq N$

$$\det(A^{(k)}) = \det(A_{1,1}) \cdot \det(S_i^{(k-i)}) \neq 0$$

et donc S_i est fortément inversible. □

LEMME 2.5. *Soit*

$$A = \begin{matrix} \begin{bmatrix} A_{1,1} & A_{1,2} \\ A_{2,1} & A_{2,2} \end{bmatrix} & \begin{matrix} i \\ N-i \end{matrix} \\ \begin{matrix} i & N-i \end{matrix} & \end{matrix}$$

tel que A et $A_{1,1}$ soient inversibles, et soit $S_i \in \mathbb{K}^{(N-i)\times(N-i)}$ le i-ème complément de Schur, alors

$$A^{-1} = \begin{bmatrix} A_{1,1}^{-1} + A_{1,1}^{-1}A_{1,2}S_i^{-1}A_{2,1}A_{1,1}^{-1} & -A_{1,1}^{-1}A_{1,2}S_i^{-1} \\ -S_i^{-1}A_{2,1}A_{1,1}^{-1} & S_i^{-1} \end{bmatrix}.$$

DÉMONSTRATION. On inverse A *via* la factorisation 11, alors

$$A^{-1} = \begin{bmatrix} A_{1,1} & A_{1,2} \\ & S_i \end{bmatrix}^{-1} \cdot \begin{bmatrix} \mathbf{1}_i & \\ A_{2,1}A_{1,1}^{-1} & \mathbf{1}_{N-i} \end{bmatrix}^{-1} = \begin{bmatrix} A_{1,1}^{-1} & -A_{1,1}^{-1}A_{1,2}S_i^{-1} \\ & S_i^{-1} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{1}_i & \\ -A_{2,1}A_{1,1}^{-1} & \mathbf{1}_{N-i} \end{bmatrix};$$

on obtient l'expression pour A^{-1} en faisant la multiplication par blocs. □

LEMME 2.6. *Soit A fortément inversible, alors pour $h, k \geq 0$:*

$$\mathcal{S}^{(h)}(\mathcal{S}^{(k)}(A)) = \mathcal{S}^{(h+k)}(A) \quad , \quad (\mathcal{S}^{(k)}(A))^{(h)} = \mathcal{S}^{(k)}(A^{(h+k)}).$$

Ces matrices correspondent aux blocs indiqués graphiquement :

DÉMONSTRATION. Le complément de Schur $\mathcal{S}^{(h)}(\mathcal{S}^{(k)}(A))$ est la matrice obtenue après h pas de l'algorithme d'élimination sur $\mathcal{S}^{(k)}(A)$; soit celle qu'on obtient après $h + k$ pas de l'algorithme d'élimination sur A, donc elle coïncide avec $\mathcal{S}^{(h+k)}(A)$. En outre, on a

$$\begin{aligned} (\mathcal{S}^{(k)}(A))^{(h)} &= (A_{2,2} - A_{2,1}A_{1,1}^{-1}A_{1,2})^{(h)} \\ &= A_{2,2}^{(h)} - A_{2,1}^{(h)}A_{1,1}^{-1}A_{1,2}^{(h)} \\ &= \mathcal{S}^{(k)}(A^{(h+k)}). \end{aligned}$$

□

Grâce au lemme 2.5, on peut calculer l'inversion d'une matrice A fortément inversible à celle du bloc $A_{1,1}$ et du complément de Schur $\mathcal{S}^{(k)}(A)$, et ce procédé peut continuer recursivement jusqu'à arriver à des blocs 1×1. En fait, le calcul se fait au sens inverse, par un procédé de remontée, qui commence avec l'inversion de $A^{(1)} = [A_{1,1}]$ et de $A_{2,2} \in \mathbb{K}^\times$, ensuite le $1\times$-complément de Schur de $A^{(2)}$, finalement on inverse $A^{(2)} \in \mathbb{K}^{2\times 2}$ *via* les formules du lemme 2.5, etc.. Pour que ce schéma soit efficace, il faut que la partition soit *balancée*, c'est-à-dire $k = \lfloor N/2 \rfloor$.

Ceci conduit à un arbre binaire. La figure ci-dessous graphique en noir les 1×1 blocs qu'on inverse directement, et en blancs ceux qu'on inverse recursivement, et donc le calcul ne fait appel qu'à des multiplications de matrices :

Algorithme d'inversion de Strassen :
Entrée : $A \in \mathbb{K}^{N\times N}$ **;**
Sortie : $A^{-1} \in \mathbb{K}^{N\times N}$ **;**

(1) Construction d'une partition balancée : soit $k = \lfloor N/2 \rfloor$ et soit

$$A = \begin{bmatrix} A_{1,1} & A_{1,2} \\ A_{2,1} & A_{2,2} \end{bmatrix} \begin{matrix} k \\ N-k \end{matrix}$$
$$\begin{matrix} k & N-k \end{matrix}$$

la partition associée ;

(2) Calcul de

$$A_{1,1}^{-1}$$

directement si $k = 1$; sinon on le fait recursivement ;

(3) Calcul du complément de Schur

$$S \leftarrow A_{2,2} - A_{2,1}A_{1,1}^{-1}A_{1,2};$$

(4) Calcul de

$$S^{-1}$$

directement si $N - k = 1$; sinon on le fait recursivement ;

(5) Calcul de A^{-1} *via* les formules du lemme 2.5.

end.

THÉORÈME 2.7. *Soit $m(N)$ la complexité de multiplier deux matrices $N \times N$, et suppossons $m(N) = O(N^\beta)$ pour un $\beta \geq 2$, alors l'algorithme d'inversion recursive calcule A^{-1} en $O(N^\beta)$ ops.*

DÉMONSTRATION. Notons $a(N) := \mathcal{C}_{Strassen}(N)$ la complexité de l'algorithme d'inversion recursive. Le calcul du complément de Schur S on calcule d'abord

$$\alpha := A_{2,1} \cdot A_{1,1}^{-1} \quad , quad\beta := \alpha \cdot A_{1,2} = A_{2,1}A_{1,1}^{-1}A_{1,2},$$

puis on obtient le complément de Schur avec une somme. Pour le calcul de A^{-1} *via* les formules du lemme 2.5 on calcule

$$\gamma := A_{1,1}^{-1} \cdot A_{1,2}, \quad \delta := S^{-1} \cdot \alpha, \quad \varepsilon := \gamma \cdot \delta, \quad \theta := \gamma \cdot S^{-1}.$$

Donc

$$a(N) \leq 2a(\lceil N/2 \rceil) + 6m(N) + O(N^2) = 2a(\lceil N/2 \rceil) + O(N^\beta).$$

Si l'on suppose par recurrence $a(\lceil N/2 \rceil) \leq c\lceil N/2 \rceil^\beta$ alors

$$a(N) \leq 2c\lceil N/2 \rceil^\beta \leq cN^\beta.$$

□

2.4. Inversion super-rapide de matrices structurées. Soit $A \in \mathbb{K}^{N \times N}$ une matrice fortement inversible et structurée. Pour calculer l'inverse en temps quasi-linéaire, on applique l'algorithme d'inversion recursive aux générateurs.

Algorithme d'inversion recursive de matrices structurées :

Entrée : $S, T \in \mathbb{K}^{N \times N}$ triangulaires inférieure et supérieure respectivement telles que $\mathrm{Spec}(S) \cap \mathrm{Spec}(T) = \emptyset$ et $G, B \in \mathbb{K}^{N \times \alpha}$ générateurs d'une matrice A fortément inversible ;

Sortie : $\widetilde{G}, \widetilde{B} \in \mathbb{K}^{N \times \alpha}$ (T, S)-générateurs de A^{-1}.

(1) Construction d'une partition balancée : soit $k = \lfloor N/2 \rfloor$ et soit

$$G = \begin{bmatrix} G_1 \\ G_2 \end{bmatrix}, \qquad B = \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} \begin{matrix} k \\ N-k \end{matrix}$$

la partition associée ;

(2) Recursivement on calcule

$$\widetilde{G}_1, \widetilde{B}_1$$

des générateurs de $\nabla_{T_{1,1},S_{1,1}}(A_{1,1}^{-1})$ **;**

(3) Calcul des $(S_{2,2}, T_{2,2})$**-générateurs de déplacement** $\nabla_{S_{2,2},T_{2,2}}(S)$ **du complément de Schur** *via* **les formules 4 ;**

(4) Recursivement on calcule des générateurs de $\nabla_{T_{2,2},S_{2,2}}(S^{-1})$ **;**

(5) Calcul de A^{-1} *via* **les formules du lemme 2.5.**

end.

Pour obtenir des algorithmes super-rapides, on est obligé de traiter les entrées et sorties en forme compacte en termes de générateurs : déjà l'écriture dense des matrices coûte N^2 ops.

Le coût de cet algorithme dépends essentiellement de la multiplication d'un vecteur avec une matrice structurée : par exemple, le calcul des générateurs pour le complément de Schur donne

$$\nabla_{S_{2,2},T_{2,2}}(A_{(k)}) = (G_2 - A_{2,1}A_{1,1}^{-1}G_1)(B_2 - B_1A_{1,1}^{-1}A_{2,1})$$

donc il faut multiplier chacune des α colonnes de G_1 par une matrice structurée.

PROPOSITION 2.8. *Le coût de l'algorithme d'inversion recursive de matrices structurées est de* $O(\alpha^2(N + m_{S,T}(N))\log(N))$.

2.5. Multiplication matrice-vecteur pour des matrices structurées. Le coût de la multiplication matrice-vecteurs pour les structures classiques de dimension N est de

$O(N\log(N))$ pour les matrices Toeplitz et Hankel et

$O(N\log^2(N))$ pour les matrices Vandermonde et Cauchy.

Ces estimations s'étendent aux matrices *type* Toeplitz, etc et leurs inverses, *via* les expressions bilinéaires des matrices en termes de générateurs.

Pour les matrices de Toeplitz, la multiplication matrice-vecteur est essentiellement une convolution, et donc se calcule en $O(N\log(N))$ ops grâce à la TFR. Faisons le cas $N = 3$: la multiplication

$$\begin{bmatrix} a_0 & a_{-1} & a_{-2} \\ a_1 & a_0 & a_{-1} \\ a_2 & a_1 & a_0 \end{bmatrix} \cdot \begin{bmatrix} v_0 \\ v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} w_0 \\ w_1 \\ w_2 \end{bmatrix}$$

se calcule comme les coefficients des termes de degré 0,1,2 dans le produit

$$(a_{-2}x^{-2} + a_{-1}x^{-1} + a_0 + a_1x + a_2x^2)(v_0 + v_1x + v_2x^2).$$

Notons que le codage consiste en une multiplication du mesagge par une matrice de Vandermonde, donc peut se faire en temps quasi-linéaire $O(N\log^2(N)\log\log(N)))$.

3. Une application aux codes correcteurs d'erreurs

Références pour cette section : [**2, 15**]. Dans la page **(adresse)** on peut jouer avec une implémentation interactive de décodage des codes de Reed-Solomon, ce qui nous permettra une première approche concrète aux notions de codification, bruit et décodage.

Dans la pratique, la transmission de l'information se fait par des canaux "bruyants" introduisant des erreurs pendant la transmission. Exemples de cette situation :

– Transmission des données *via* satellite ;

– Stockage d'information sur un support (données numériques, musique, images) genre cassette ou CD, pour sa récuperation ultérieure ;
– Transmission d'information entre des ordinateurs (Internet).

La solution est de *coder* le message envoyé, a fin de pouvoir détecter et corriger les erreurs produits au cours de la transmission. On étudiera des codes où le *message* ou *mot* à envoyer est une suite de longueur k de symboles d'un alphabet fixé L. Ces messages sont codés avant la transmission, le *message (ou mot) codé* sera une suite de longueur n de symboles du même alphabet L. On a besoin d'une certaine redondance pour que le message originel soit récupérable, donc on suppose $n \geq k$.

En théorie algébrique de codes, l'alphabet de transmission forme un corps fini

$$L = \mathbb{F}_q \qquad (q = p^r).$$

Vue la construction des ordinateurs, il est naturel de considérer un alphabet $\mathbb{F}_2 = \{0, 1\}$ ou encore $\mathbb{F}_{2^r} = \{0, 1\}^r$, toutefois les constructions qu'on fera seront valables pour un alphabet $\mathbb{F}_q$ pour $q = p^r$ quelconque. La *codification* est une fonction injective

$$E : \mathbb{F}_q^k \hookrightarrow \mathbb{F}_q^n.$$

On se restreindra au cas des codes *linéaires*, dont la codification E est une fonction linéaire. Dans ce cas, on notera aussi $[E] \in \mathbb{F}^{n\times k}$ la *matrice génératrice* de la codification. Souvent une codification est de la forme $[E] = [\mathbf{1}_k|P]$ où P est une matrice de taille $k \times (n - k)$. Si l'on pose $Ex = y$, pour $1 \leq i \leq k$ les y_i sont les *positions d'information*, et pour $k + 1 \leq i \leq n$ sont les *parity checks*. Ceci se révèle utile lorsqu'il n'y a pas eu d'erreurs dans la transmission, car alors on peut récupérer le message à partir des positions d'information, sans avoir à le décoder.

L'ensemble de mots codés

$$\mathcal{C} := E(\mathbb{F}_q^k) \subset \mathbb{F}_q^n$$

est par définition le *code*, c'est un sous-espace vectoriel de $\mathbb{F}_q$ de dimension k. Le *décodage* est une rétraction de E, c'est-à-dire une fonction $D : \mathbb{F}_q^n \to \mathbb{F}_q^k$ telle que $D \circ E = \mathbf{1}_{\mathbb{F}_q^k}$.

La *distance de Hamming* entre deux mots $v, w \in \mathbb{F}_q^n$ est

$$\operatorname{dist}(v, w) := \operatorname{Card}\{i \ : \ v_i \neq w_i\}.$$

Pour $v \in \mathbb{F}_q^n$ on désigne $B_\rho(v)$ la boule de rayon ρ (par rapport à la distance de Hammings) et centrée en v :

$$B_\rho(v) := \{w \in \mathbb{F}_q^n \ : \ \operatorname{dist}(v, w) \leq \rho\}.$$

Autrement-dit, c'est l'ensemble des w différant de v en au plus ρ coordonnées. La *distance minimale* de $\mathcal{C}$ est

$$\begin{aligned}\operatorname{dist}(\mathcal{C}) &:= \min\{\operatorname{dist}(v, w) \ : \ v, w \in \mathcal{C}, v \neq w\} \\ &= \min\Big\{ \operatorname{dist}(v, 0) \ : \ v \in \mathcal{C} \setminus \{0\}\Big\} \\ &= \min\Big\{\operatorname{Card}\{i \ : \ v_i \neq 0\} : v \in \mathcal{C} \setminus \{0\}\Big\},\end{aligned}$$

la deuxième égalité étant une conséquence de la linéarité. Un code $\mathcal{C}$ sur un alphabet $\mathbb{F}_q$, de *dimension* (longueur du message) k, *longueur* (du mot codée) n et distance minimale d, est appelé un $[n, k, d]_q$*-code*.

Soit $y \in \mathbb{F}_q^n$ tel que $\operatorname{dist}(y, v) \leq d - 1$ pour un certain $v \in \mathcal{C}$, alors $y \in \mathcal{C}$ si et seulement si $y = v$. De plus, si $d = 2\,t + 1$ et $\operatorname{dist}(y, v) \leq t$, alors v est l'unique élément $w \in \mathcal{C}$ tel que $\operatorname{dist}(y, w) \leq t$. On en déduit la proposition suivante :

PROPOSITION 3.1. *Soit $\mathcal{C}$ un code à distance minimale d, alors $\mathcal{C}$ peut détecter jusqu'à $d-1$ erreurs. Si $d \geq 2t+1$, alors $\mathcal{C}$ peut corriger jusqu'à t erreurs.*

Comme exemple, considérons les *codes de répétition* :

- Deux répétitions :
$$(a,b,c) \mapsto (a,b,c,a,b,c).$$
On vérifie aisément que c'est un $[6,3,2]_q$; donc il détecte jusqu'à un erreur, mais il ne peut pas le corriger.
- Trois répétitions :
$$(a,b,c) \mapsto (a,b,c,a,b,c,a,b,c)$$
. C'est un $[(9,3,3]_q$; donc il détecte 2 erreurs et il corrige 1.

Faisons l'analyse des codes de répétition. Disons qu'on veut passer un message de longueur k sur un alphabet $\mathbb{F}_q$; la solution bêtement proposé par ces codes est de le répéter ℓ fois, c'est-à-dire
$$(a_1,\dots,a_k) \mapsto (\underbrace{a_1,\dots,a_k,a_1,\dots,a_k,a_1,\dots,a_k}_{k\,\ell}).$$
C'est un $[k\,\ell,k,\ell]_q$-code, et donc il peut détecter jusqu'à ℓ erreurs et corriger $\lfloor(\ell-1)/2\rfloor$. Le quotient
$$\frac{d}{n} = \frac{\ell}{k\,\ell} = \frac{1}{k}$$
est constant par rapport au nombre ℓ des répétitions, et beaucoup trop petit pour être utile en pratique. Les "bons" codes sont ceux pour lesquels le *quotient d'information* k/n n'est pas trop petit, et qu'en même temps d est grand.

Notons finalement que pour un code linéaire, la codification se réduit à la multiplication matrice-vecteur, donc prends au plus $O(k\,n)$ ops de $\mathbb{F}_q$.

Voici quelques exercices sur la structure des corps finis. La notation $\mathbb{F}_p$ désigne le corps $\mathbb{Z}/p\mathbb{Z}$, p étant un nombre premier. On admettra que tout corps (commutatif) possède une clôture algébrique, et que deux clôtures algébriques d'un même corps sont isomorphes.

EXERCICE 5.2. ◁ Soit k un corps quelconque, et soit $g \in k[X]$ un polynôme *irréductible*, c'est à dire polynôme non constant qui n'est pas le produit de deux autres polynômes non constants. Montrer que l'idéal $(g) \subset k[X]$ engendré par g est maximal. En déduire que $k[X]/(g)$ est un corps.

Indication : utiliser le théorème de Bézout qui énonce que si $g,h \in k[X]$ sont premiers entre eux, alors ils existent $u,v \in k[X]$ tels que
$$gu + hv = 1.$$

▷

EXERCICE 5.3. ◁

(i) Montrer que $g = x^4 + x + 1 \in \mathbb{F}_2[x]$ est irréductible.

(ii) Combien d'éléments y a-t-il dans $\mathbb{F} := \mathbb{F}_2[x]/(g)$?

(iii) Notons $\overline{x}$ la classe de x dans le corps quotient $\mathbb{F}$. Calculer les différentes puissances de $\overline{x}$. Vérifier que $1,\overline{x},\overline{x}^2,\overline{x}^3$ est une base de $\mathbb{F}$ comme espace vectoriel sur $\mathbb{F}_2$.

(iv) Montrer que $\{0,1,\overline{x}^5,\overline{x}^{10}\}$ est un sous-corps de $\mathbb{F}$ à quatre éléments.

(v) Y a-t-il un sous-corps de $\mathbb{F}$ à huit éléments ? Y a-t-il d'autres sous-corps ?

▷

EXERCICE 5.4. ◁ Soit $\mathbb{F}$ un corps fini. Montrer qu'il existe un nombre premier p et un entier n tel que $\mathrm{Card}\mathbb{F} = p^n$. Indication : montrer que le sous corps engendré par 1 est un $\mathbb{F}_p$ et que $\mathbb{F}$ est un espace vectoriel de dimension finie sur ce corps. ▷

EXERCICE 5.5. ◁ Notons $\overline{\mathbb{F}_p}$ la clôture algébrique de $\mathbb{F}_p$, et soit n un entier quelconque. On pose $q = p^n$, et on considère l'ensemble

$$Z_q = \{x \in \overline{\mathbb{F}_p} : x^q = x\}.$$

Montrer que $\mathrm{Card}Z_q = q$, et que Z_q est un corps. Indication : montrer que $f(X) = X^q - X$ est un polynôme sans racines multiples. ▷

EXERCICE 5.6. ◁ Soit $\mathbb{F}$ un corps fini contenu dans $\overline{\mathbb{F}_p}$, de cardinal $q = p^n$. Montrer que tout $x \in \mathbb{F}^\times$ vérifie $x^{q-1} = 1$. En conclure que $\mathbb{F} = Z_q$. ▷

EXERCICE 5.7. ◁ Soit $\mathbb{F}_{p^n}$ le seul sous-corps de $\overline{\mathbb{F}_p}$ à p^n éléments. Montrer que

$$\mathbb{F}_p = \bigcup_{n \geq 1} \mathbb{F}_{p^n}.$$

Indication : utilizer que pour $\xi \in \overline{\mathbb{F}_p}$, le corps engendré $\mathbb{F}_p(\xi)$ est fini. ▷

EXERCICE 5.8. ◁ Dans cet exercice on donne plusieurs bornes pour les paramètres associés aux codes. Une façon de produire des bons codes est de fixer une longueur n et une distance minimale d, puis d'essayer de maximiser k en prenant les mots du code une par une, tout en gardant $\mathrm{dist}(v, w) \geq d$.

(1) Montrer que pour tout $c \in \mathbb{F}_q^n$

$$b(n, d) := \mathrm{Card}(B_{d-1}(c)) = \sum_{i=0}^{d-1} \binom{n}{i} (q-1)^i.$$

(2) Soit $d \geq 1$ et $\mathcal{C} \subset \mathbb{F}_q^n$ un sous-ensemble (pas forcément un sous-espace linéaire) tel que $\mathrm{dist}(v, w) \geq d$ pour tout $v \neq w$ dans avec $v, w \in \mathcal{C}$. Supposons que pour tout $z \in \mathbb{F}_q^n \setminus \mathcal{C}$ on a $\mathrm{dist}(z, c) \leq d - 1$ pour un certain $c \in \mathcal{C}$. Montrer que

$$b(n, d) \cdot \mathrm{Card}(\mathcal{C}) \geq q^n.$$

Ceci est la *estimation de Gilbert-Varshamov*. Indication : De façon équivalente, montrer que si $b(n, d) \cdot \mathrm{Card}(\mathcal{C}) < q^n$ alors il existe z tel que la distance minimale de $\mathcal{C} \cup \{z\}$ est encore $\geq d$.

(3) Montrer que si

$$b8n, d) < q^{n-k+1}$$

pour un certain k, alors il existe un $[n, k, d]_q$-code linéaire. Indication : par récurrence, on peut supposer qu'il existe un $[n, k-1, d]_q$-code linéaire $\mathcal{C}$. En appliquant la partie (2), considérez le code linéaire $\mathcal{C}'$ engendré par $\mathcal{C}$ et z, où la distance de z à n'importe quelle mot de $\mathcal{C}$ est $\geq d$, et montrer que $\mathcal{C}'$ a encore distance minimale d.

(4) Dans l'autre direction, montrer que pour tout code linéaire on a

$$d \leq n - k + 1.$$

Ceci est la *Singleton bound*. Indication : considérez ce qu'il se passe quand un sous-ensemble de $d - 1$ coordonnées est effacé de chacune des mots du code.

▷

3.1. Codes de Reed-Solomon. Pour $q = p^r$ et $k \leq n \leq q-1$ considérons une suite de points deux à deux distincts

$$x_1, \ldots, x_n \in \mathbb{F}_q^\times.$$

On identifie $\mathbb{F}_q^k$ avec $\mathbb{F}_q[x]_{\leq k-1}$, l'espace vectoriel des polynômes sur $\mathbb{F}_q$ de degré au plus $k-1$ et considérons l'application linéaire injective

$$E : \mathbb{F}_q[x]_{\leq k-1} \to \mathbb{F}_q^n \quad , \quad f \mapsto (f(x_1), \ldots, f(x_n)).$$

On pose

$$\mathrm{RS}(k, q; x_1, \ldots, x_n) := E(\mathbb{F}_q[x]_{\leq k-1})$$

le *code de Reed-Solomon* sur $\mathbb{F}_q$ de dimension k et longueur n associé aux points $x_1, \ldots, x_n$. Un polynôme non nul f de degré au plus $k-1$ ne peut pas avoir plus de $k-1$ zéros, et donc

$$\mathrm{dist}(v, 0) \geq n - k + 1 \quad \text{pour toute mot } v \in \mathcal{C} \setminus \{0\}.$$

Comme on peut construire $f \in \mathbb{F}_q[x]_{\leq k-1}$ avec $k-1$ zéros parmi les x_is, on a

$$\mathrm{dist}(\mathrm{RS}(k, q; x_1, \ldots, x_n)) = n - k + 1.$$

La *singleton bound* (exercice 5.8(d)) montre que les codes de Reed-Solomon ont la distance minimale d la plus grande possible, parmi tout les codes de longueur n et dimension k. Les codes avec cette propriété s'appellent *codes séparables à distance maximale* (en anglais : *maximum distance separable codes*) dans la littérature.

Exercice 5.9. ◁

(1) Montrer que

$$g = x^3 + x + 1$$

est irréductible sur $\mathbb{F}_2$. En déduire que

$$\mathbb{F}_8 = \mathbb{F}_2[x]/g,$$

et que $1, \overline{x}, \overline{x}^2$ est une base de $\mathbb{F}_8$ comme $\mathbb{F}_2$-espace linéaire.

(2) Énumérer le code de Reed-Solomon sur $\mathbb{F}_8$ de longueur 4 et dimension 2, associé aux éléments

$$x_1 := 1, \quad x_2 := 1 + \overline{x}, \quad x_3 := 1 + \overline{x} + \overline{x}^2, \quad x_4 := 1 + \overline{x}^2.$$

▷

Exercice 5.10. ◁ Soit $\mathcal{C}$ une code de Reed-Solomon de longueur n et dimension k, sur un alphabet $\mathbb{F}_{2^r}$. Ainsi chaque mot codée peut se représenter comme une suite de $r\,n$ bits, car chaque symbole de $\mathbb{F}_{2^r}$ est représenté par r bits.

Montrer qu'une suite de $r\,\ell$ erreurs consécutifs à niveau bit, change au plus $\ell + 1$ des symboles d'un mot codée, à niveau $\mathbb{F}_{2^r}$. En déduire que si

$$\ell + 1 \leq \lfloor (n-k)/2 \rfloor,$$

le code $\mathcal{C}$ peut corriger une suite de $r\,\ell$ erreurs consécutifs. ▷

3.2. Décodage des codes de Reed–Solomon. Référence pour cette section : [**12**]. Le décodage des codes de Reed–Solomon se fait *via* l'algorithme de Berlekamp et Massey. Considérons un alphabet $\mathbb{F}_q$ ($q = p^r$) et soit m un message de longueur k qu'on idéntifie à un polynôme de degré borné par $k-1$:

$$f = m_1 + m_2\,x + \cdots + m_k\,x^{k-1} \in \mathbb{F}_q[x]_{k-1}.$$

Soient $x_1, \ldots, x_n \in \mathbb{F}_q^\times$ des points distincts et soit

$$E(f) := (f(x_1), f(x_2), \ldots, f(x_n)) \in \mathbb{F}_q^n$$

et $t(E(m)) = (t_1, \dots, t_n)$ le message reçu. Notre but est alors de récuperer f comme l'unique élément de $\mathbb{F}_q[x]_{k-1}$ tel que

$$\operatorname{dist}(t, E(f)) \leq \frac{d-1}{2} = \frac{n-k}{2}.$$

Autrement-dit, f est l'unique polynôme dans $\mathbb{F}_q[x]_{k-1}$ tel que $f(x_i) = t_i$ pour au moins $n - \dfrac{n-k}{2} = \dfrac{n+k}{2}$ des x_is.

PROPOSITION 3.2. *Soit $k \equiv n \pmod 2$ et soient g, h des polynômes tels que $h \neq 0$,*

$$\deg(g) < \frac{n+k}{2} \quad , \qquad \deg(h) \leq \frac{n-k}{2},$$

et tels que $g(x_i) + t_i\, h(x_i) = 0$ pour $i = 1, \dots, n$, alors $f = -g/h$.

Notons que g, h existent puisqu'il s'agit d'un système de n équations linéaires en $n+1$ variables.

DÉMONSTRATION. Posons $H := g + f \cdot h$, alors

$$\deg(H) \leq \max(\deg(g), \deg(f\,h)) \leq \max\left(\frac{n+k}{2} - 1, k - 1 + \frac{n-k}{2}\right) \leq \frac{n+k}{2} - 1.$$

En outre

$$\operatorname{Card}\{\xi \in \mathbb{F}_q \,:\, H(\xi) = 0\} \geq \operatorname{Card}\{x_i \,:\, f(x_i) = t_i\} \geq \frac{n+k}{2}$$

et donc $H \equiv 0$, c'est-à-dire $f = -g/h$. □

Le système linéaire associé est

$$\mathrm{BM} \cdot \begin{bmatrix} g^T \\ h^T \end{bmatrix} = \begin{bmatrix} 1 & x_1 & \cdots & x_1^{(n+k)/2-1} & y_1 & y_1\,x_1 & \cdots & y_1\,x_1^{(n-k)/2} \\ \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \\ 1 & x_n & \cdots & x_n^{(n+k)/2-1} & y_n & y_n\,x_n & \cdots & y_n\,x_n^{(n-k)/2} \end{bmatrix} \cdot \begin{bmatrix} g_0 \\ \vdots \\ h_{(n-k)/2} \end{bmatrix}.$$

La matrice $\mathrm{BM} \in \mathbb{F}_q^{n \times (n+1)}$ n'est pas une matrice de Vandermonde mais presque ! Posons

$$S := \operatorname{diag}(x_1^{-1}, \dots, x_n^{-1}) \in \mathbb{F}_q^{n \times n} \quad , \quad T := Z_0 \in \mathbb{F}_q^{(n+1)\times(n+1)}$$

on vérifie

$$\nabla_{S,T}(\mathrm{BM}) = \begin{bmatrix} x_1^{-1} & 0 & \cdots & 0 & (y_1\,x_1^{-1} - x_1^{(n+k)/2-1}) & 0 & \cdots & 0 \\ \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \\ x_n^{-1} & 0 & \cdots & 0 & (y_n\,x_n^{-1} - x_n^{(n+k)/2-1}) & 0 & \cdots & 0 \end{bmatrix} \tag{12}$$

qui est une matrice de rang 2. Le calcul du couple (g, h) revient à trouver un élément non nul dans le noyau de la matrice BM, ce qu'on fait en calculant la décomposition $\mathrm{BM} = P \cdot L \cdot U$ car $\ker(\mathrm{BM}) = \ker(U)$. La résolution consiste en trois étapes :

(1) décomposition $\mathrm{BM} = P^{(n\times n)}\, L^{(n\times n)}\, U^{(n\times(n+1))}$;
(2) détermination de $(g, h) \in \ker(U) \setminus \{0\}$;
(3) division de polynômes $f = -g/h$.

La partie la plus coûteuse est le calcul de la décomposition PLU. Pour cela, on tirera profit du fait que BM est structurée : avec un algorithme rapide, cette décomposition se fait en

$$O(\operatorname{rang}_{S,T}(\mathrm{BM})\, n^2) = O(n^2) \quad \text{ops de } \mathbb{F}_q.$$

Le calcul du noyau de $U \in \mathbb{F}_q^{n\times(n+1)}$ se fait par *backward substitution* en n^2 ops, puisqu'il s'agit d'un système triangulaire. Finalement, on a $\deg(g), \deg(h) = O(n)$ donc la division g/h se fait en $O(n^2)$ ops par l'algorithme standard, ou en $O(n\log(n))$ ops par l'analogue polynomial de l'algorithme de Schonhäge-Strassen. Le coût total reste en

$$O(n^2) \quad \text{ops de } \mathbb{F}_q.$$

On a implémenté la méthode sur `Maple9` pour la tester sur un exemple petit, *voir* le fichier `ReedSolomon.mw` joint. L'exemple considéré (un message de longueur 4 et dimension 2 sur l'alphabet $\mathbb{F}_8$) est instructif et on le décrira avec un certain détail.

Le polynôme $x^3 + x + 1 \in \mathbb{F}_2[x]$ est irréductible donc

$$\mathbb{F}_8 = \mathbb{F}_2[x]/(x^3 + x + 1).$$

Posons $a := \overline{x}$; tout élément $b \in \mathbb{F}_8$ s'écrit alors comme

$$b = b_0 + b_1 a + b_2 a^2 \qquad \text{avec } b_i = 0, 1.$$

Le message à envoyer est $m = (1, 1)$, et on le représente comme le polynôme $f = 1 + x \in \mathbb{F}_8[x]$. Prenons quatre éléments témoins dans $\mathbb{F}_8^\times$:

$$x_1 = 1, \quad x_2 = 1 + a, \quad x_3 = 1 + a + a^2, \quad x_4 = 1 + a^2,$$

et disons que le mot reçu est

$$y_1 = 0, \quad y_2 = a, \quad y_3 = a, \quad y_4 = a^2.$$

Il y a un seul erreur a niveau $\mathbb{F}_8$: $y_3 \neq f(x_3)$, et notre code de Reed-Solomon peut le corriger car $(n - k)/2 = 1$. La matrice de Berlekamp-Massey associée à ces x_is et y_js est

$$\begin{aligned}\mathrm{BM} &= \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 1+a & (1+a)^2 & a & a(1+a) \\ 1 & 1+a+a^2 & (1+a+a^2)^2 & a & a(1+a+a^2) \\ 1 & 1+a^2 & (1+a^2)^2 & a^2 & a^2(1+a^2 \end{bmatrix} \\ &= \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 1+a & 1+a^2 & a & a+a^2 \\ 1 & 1+a+a^2 & 1+a & a & 1+a^2 \\ 1 & 1+a^2 & 1+a+a^2 & a^2 & a \end{bmatrix}.\end{aligned}$$

On calculera se décomposition LU *via* l'algorithme rapide. pour vérification ultérieure, le résultat est

$$L = \begin{bmatrix} 1 & & & \\ 1 & 1 & & \\ 1 & 1+a & 1 & \\ 1 & a & 1 & 1 \end{bmatrix}, \quad U = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ & a & a^2 & a & a+a^2 \\ & & 1+a^2 & a^2 & a^2 \\ & & & a^2 & 1 \end{bmatrix}.$$

Avec ceci on calcule $\ker(\mathrm{BM}) = \ker(U)$; ce noyau est engendré par le vecteur

$$(v_1, \ldots, v_5) = (1 + a + a^2, a^2 + a, 1, 1 + a + a^2, 1) \in \mathbb{F}_8^5.$$

On construit les polynômes de Berlekamp-Massey associés

$$g = v_1 + v_2 x + v_3 x^2 = 1 + a + a^2 + (a^2 + a)x + x^2, \quad h = v_4 + v_5 x = 1 + a + a^2 + x;$$

le message reconstruit est donc $f = -g/h = 1 + x$.

C'est très bien, mais ce qui nous intéresse c'est de voir marcher l'algorithme rapide ! Les matrices de déplacement sont

$$S = \mathrm{diag}(x_1^{-1}, \ldots, x_4^{-1}) = \begin{bmatrix} 1 & & & \\ & a+a^2 & & \\ & & a^2 & \\ & & & a \end{bmatrix}$$

et $T = Z_0$ un bloc 5×5 de Jordan. On calcule des générateurs pour le déplacement de BM à l'aide de la formule (12), sans avoir à expliciter BM :

$$G(1) = \begin{bmatrix} 1 & 1 \\ a+a^2 & a \\ a^2 & 0 \\ a & a^2 \end{bmatrix}, \quad B(1) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}.$$

On construit progressivement la matrice $U = [U_{i,j}]_{1 \le i \le 4, 1 \le j \le 5}$. On devra calculer aussi L, puisqu'on en a besoin pour l'actualisation des générateurs du déplacement des compléments de Schur succéssifs. Suivant l'algorithme rapide, à chaque étape on construit une ligne de U, puis on actualise les générateurs. La première ligne de U coïncide avec celle de BM :

$$U_{1,1} = 1, \quad U_{1,2} = 1, \quad U_{1,3} = 1, \quad U_{1,4} = 0, \quad U_{1,5} = 0.$$

Puis on calcule des générateurs pour BM(2), le déplacement du premier complément de Schur, *via* les formules (4) :

$$G(2) = \begin{bmatrix} 1+a+a^2 & 1+a \\ 1+a^2 & 1 \\ 1+a & 1+a^2 \end{bmatrix}, \quad B(2) = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$

Passons à la deuxième itération de la boucle. Maintenant on veut calculer la deuxième ligne de U. Pour cela il faut reconstruire la première ligne de BM(2), donc on calcule la première ligne de son déplacement :

$$\nabla(BM(2)) = \begin{bmatrix} 1+a+a^2 & 1+a+a^2 & 1+a & 0 \\ * & * & * & * \\ * & * & * & * \end{bmatrix}$$

puis on reconstruit la première ligne de BM(2) à l'aide des formules du lemme (2.2) :

$$\mathrm{BM}(2) = \begin{bmatrix} a & a^2 & a & a+a^2 \\ * & * & * & * \\ * & * & * & * \end{bmatrix}.$$

Ainsi la deuxième ligne de U est

$$U_{2,1} = 0, \quad U_{2,2} = a, \quad U_{2,3} = a^2, \quad U_{2,4} = a, \quad U_{2,5} = a + a^2.$$

L'actualisation des générateurs donne

$$G(3) = \begin{bmatrix} 1+a+a^2 & a^2 \\ a+a^2 & 1+a \end{bmatrix}, \quad B(3) = \begin{bmatrix} 1+a & 1 & 1+a \\ 0 & 1 & 0 \end{bmatrix}.$$

Et c'est reparti ! La première ligne du déplacement de BM(3) est

$$\nabla(BM(3)) = \begin{bmatrix} a & 1+a & a \\ * & * & * \end{bmatrix}$$

et donc $\mathrm{BM}(3) = \begin{bmatrix} 1+a^2 & a^2 & a^2 \\ * & * & * \end{bmatrix}$, alors la troisième ligne de U est

$$U_{3,1} = 0, \quad U_{3,2} = 0, \quad U_{3,3} = 1 + a^2, \quad U_{3,4} = a^2, \quad U_{3,5} = a^2.$$

L'actualisation des générateurs donne

$$G(4) = \begin{bmatrix} 1 & 1+a+a^2 \end{bmatrix}, \quad B(4) = \begin{bmatrix} a^2 & a+a^2 \\ 1 & 0 \end{bmatrix}.$$

Pour finir, on calcule encore la première ligne du déplacement de BM(4)

$$\nabla(BM(4)) = \begin{bmatrix} 1+a & a+a^2 \end{bmatrix}$$

puis BM(4) $= \begin{bmatrix} a^2 & 1 \end{bmatrix}$, la dernière ligne de U est donc

$$U_{4,1} = 0, \quad U_{4,2} = 0, \quad U_{4,3} = 0, \quad U_{4,4} = a^2, \quad U_{3,5} = 1;$$

et avec ceci on a fini le calcul.

4. Exercices

EXERCICE 5.11. ◁ Soient

$$d_0, \dots, d_\ell; n \in \mathbb{N} \quad , \quad d = d_0 + \cdots + d_\ell \quad , \quad x_1, \dots, x_n \in \mathbb{K}^\times$$

et posons

$$A := \begin{bmatrix} 1 & x_1 & \cdots & x_1^{d_0-1} & y_1 & y_1x_1 & \cdots & y_1x_1^{d_1-1} & \cdots & y_1^\ell & y_1^\ell x_1 & \cdots & y_1^\ell x_1^{d_\ell-1} \\ \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots & & \vdots & \vdots & & \vdots \\ 1 & x_n & \cdots & x_n^{d_0-1} & y_n & y_nx_n & \cdots & y_nx_n^{d_1-1} & \cdots & y_n^\ell & y_n^\ell x_n & \cdots & y_n^\ell x_n^{d_\ell-1} \end{bmatrix} \in \mathbb{K}^{n\times d}.$$

Trouvez des déplacements $S \in \mathbb{K}^{n\times n}$ et $T \in \mathbb{K}^{d\times d}$ à spectre disjoint, tels que

$$\nabla_{S,T}(A) = S \cdot A - A \cdot T$$

soit de rang au plus $\ell + 1$, et déterminez des générateurs pour $\nabla_{S,T}(A)$. ▷

4.1. Matrices de Toeplitz infinies et semi-infinies. Soit $\tau := (t_k)_{k\in\mathbb{Z}}$ une sucession de nombres complexes et posons

$$T_t := [t_{i-j}]_{i,j\in\mathbb{Z}} = \begin{bmatrix} \ddots & \ddots & & \\ \ddots & t_0 & t_{-1} & \\ & t_1 & t_0 & \ddots \\ & & \ddots & \ddots \end{bmatrix}$$

la matrice de Toeplitz bi-infinie et

$$\tau(z) := \sum_{k\in\mathbb{Z}} t_k z^k$$

le symbole associés. On suppossera $\tau(z)$ convergente dans un petit anneau autour du circle unité.

EXERCICE 5.12. ◁ Soient $\sigma := (s_k)_{k\in\mathbb{Z}}$ et $\tau := (t_k)_{k\in\mathbb{Z}}$, montrer que

(1) $T_\sigma + T_\tau = T_{\sigma+\tau}$;
(2) $\lambda \cdot T_\tau = T_{\lambda\cdot\tau}$;
(3) $T_\sigma \cdot T_\tau = T_{\sigma\cdot\tau}$;
(4) si $\tau(z) \neq 0$ pour tout $z \in S^1$ alors $T_\tau^{-1} = T_{\tau^{-1}}$.

▷

EXERCICE 5.13. ◁ Montrer qu'on peut factoriser le symbole

$$\tau(z) = \ell(z) \cdot u(z)$$

avec

$$\ell(z) = 1 + \sum_{k \geq 1} \ell_k z^k \quad , \quad u(z) = \sum_{k \leq 0} u_k z^k$$

respectivement analytique dans le disque unité et analytique dehors le disque unité. En déduire

$$T_\tau = T_\ell \cdot T_u,$$

en particulier la décomposition LU d'une Toeplitz bi-infinie est Toeplitz. ▷

EXERCICE 5.14. ◁ Montrer que T_τ n'a pas des valeurs propres, et que

$$\mathrm{Spec}(T_\tau) = \{\tau(z) : z \in S^1\}.$$

▷

EXERCICE 5.15. ◁ Montrer que $||T_\tau||_2 = |\tau(z)|_\infty$. ▷

EXERCICE 5.16. ◁ Théorème de Szegö-Grenander : suppossont T_τ normale, c'est-à-dire $T^*T = TT^*$. Soit

$$T_N := [\tau_{i-j}]_{1 \leq i,j \leq N} \in \mathbb{C}^{N \times N}$$

et considérons la mésure discrète

$$\mu_N(z) = \frac{1}{N} \sum_{\lambda \in \mathrm{Spec}(T_N)} \delta(z - \lambda).$$

Montrer que μ_N converge faiblement vers la mésure supporté et equidistribuée sur $\mathrm{Spec}(T_\tau)$. ▷

Le but des exercices suivants est d'étudier les matrices de Toeplitz semi-infinies ; en particulier on établira le celebre théorème de l'indice de Gohberg et ..., antécedant du théorème de l'indice de Atiyah et Singer.

EXERCICE 5.17. ◁On identifie le cercle unité à $\mathbb{R}/2\pi\mathbb{Z}$. Notons $C^0(S^1)$ l'ensemble des fonctions complexes continues, périodiques de période 2π. On suppose $a \in C^0(S^1)$ et on définit un opérateur de multiplication M_a dans $L^2(S^1)$ par

$$M_a u = au.$$

On rappelle que le spectre d'un opérateur A d'un espace de Hilbert dans lui même est l'ensemble des $z \in \mathbb{C}$ tels que $z - A$ possède un inverse partout défini et continu. Montrer que le spectre de M_a est exactement l'image de a. ▷

EXERCICE 5.18. ◁ Soit $b \in C^0(S^1)$; calculer $M_a M_b$. ▷

EXERCICE 5.19. ◁ On note $\hat{a}_j$ le j-ième coefficient de Fourier de a :

$$\widehat{a}_j = \int_0^{2\pi} a(\theta) e^{-\mathrm{i}j\theta} \, \mathrm{d}\theta.$$

Montrer que dans la base de Fourier formée des $e_k : \theta \mapsto \exp(\mathrm{i}k\theta)$, pour $k \in \mathbb{Z}$, l'opérateur M_a se met sous forme de matrice infinie, qu'on donnera explicitement en fonction des $\hat{a}_j$. ▷

EXERCICE 5.20. ◁ On appelle $H^2(S^1)$ l'ensemble des fonctions dans $L^2(S^1)$ dont tous les coefficients de Fourier d'indice négatif sont nuls. Vérifier que $H^2(S^1)$ est un sous-espace fermé de $L^2(S^1)$; en déduire que c'est un espace de Hilbert pour le produit scalaire induit par celui de $L^2(S^1)$. L'espace $H^2(S^1)$ est un espace de Hardy ; attention, ce n'est *pas* un espace de Sobolev. ▷

EXERCICE 5.21. ◁ On note P la projection orthogonale de $L^2(S^1)$ sur $H^2(S^1)$. Soit $a \in C^0(S^1)$; on définit un opérateur T_a dans $H^2(S^1)$ par

$$T_a u = P(M_a u).$$

Montrer que T_a est un opérateur borné de $H^2(S^1)$ dans lui-même. Donner sa matrice dans la base de Fourier. ▷

EXERCICE 5.22. ◁Soit b dans $C^0(S^1)$; est ce que, en général, T_a et T_b commutent ? ▷

EXERCICE 5.23. ◁ On note $H^2_-(S^1)$ l'ensemble des fonctions de $L^2(S^1)$ dont les coefficients de Fourier d'indice positif ou nul sont nuls, et Q la projection orthogonale de $L^2(S^1)$ sur $H^2_-(S^1)$. On définit l'application linéaire J de $L^2(S^1)$ par son expression dans la base de Fourier :

$$(\widehat{Jx})_k = \hat{x}_{-k-1}.$$

Trouver la représentation dans la base de Fourier des opérateurs

$$H_a = PM_aQJ\mid_{H^2(S^1)} \text{ et } \widetilde{H}_a = JQM_aP\mid_{H^2(S^1)}.$$

Montrer que H_a et $\widetilde{H}_a$ sont des opérateurs bornés, pour $a \in C^0(S^1)$. ▷

EXERCICE 5.24. ◁Soit a dans $C^0(S^1)$. Montrer que H_a est un opérateur compact.

Indication : comme a est une fonction continue, et que les polynômes trigonométriques sont denses dans $C^0(S^1)$, on considère une suite de polynômes trigonométriques a_n convergeant uniformément vers a ; on montrera alors que la suite des H_{a_n} converge en norme d'opérateur vers H_a et que les H_{a_n} sont de rang fini. ▷

EXERCICE 5.25. ◁ Soient a et b dans $C^0(S^1)$. Montrer que $T_aT_b - T_{ab}$ est compact.

Indication : montrer l'identité

$$T_{ab} = T_aT_b + H_a\widetilde{H}_b.$$

▷

EXERCICE 5.26. ◁ Calculer le spectre de T_{e_1}.

Indication : si $|z| \leq 1$, calculer la solution de l'équation $zx - T_ax = e_0$, et montrer qu'elle n'est pas dans $\ell^2(\mathbb{N})$. Si $|z| > 1$, calculer la solution de l'équation $zx - T_ax = y$, avec y quelconque dans $\ell^2(\mathbb{N})$; on pourra vérifier que la série

$$\sum_{j=0}^{\infty} \frac{k}{|z|^{2k}}$$

converge. ▷

EXERCICE 5.27. ◁ On considère une troncation de dimension finie n de l'opérateur T_{e_1} de la question 5.26 : c'est le bloc $n \times n$ en haut à gauche de la matrice infinie de T_{e_1} dans la base de Fourier, on le note S_n. Le ε-pseudospectre de S_n est l'ensemble des z dans $\mathbb{C}$ tels que $z - S_n$ n'est pas inversible, ou la norme de $(z - S_n)^{-1}$ est supérieure à $1/\varepsilon$. Montrer que le ε-pseudospectre de S_n est inclus dans un disque de centre 0 et de rayon $R_{n,\varepsilon}$ et contient un disque de centre 0 et de rayon $r_{n,\varepsilon}$ et que ces deux rayons tendent vers 1 quand n tend vers l'infini.

Indication : utiliser une norme $\| \|_\infty$ et le résultat de comparaison entre cette norme et la norme $\| \|_2$. ▷

EXERCICE 5.28. ◁ On note $n(A)$ la dimension du **n**oyau d'un opérateur A et $m(A)$ la dimension de l'orthogonal de son i**m**age ; un opérateur est dit de Fredholm si son noyau est de dimension finie et son image est fermée et de codimension finie ; dans ce cas, son indice est défini par

$$\mathrm{ind}(A) = n(A) - m(A).$$

Montrer que l'indice de T_{e_m} est égal à $-m$. ▷

EXERCICE 5.29. ◁ Soit A un opérateur de Fredholm ; montrer que si E est de norme assez petite, alors $A+E$ est encore de Fredholm, et son indice est le même que celui de A. ▷

EXERCICE 5.30. ◁ Soit a dans $C^0(S^1)$; on suppose que l'image de a ne contient pas 0 ; on rappelle que l'indice du chemin a par rapport à z est l'intégrale

$$\mathrm{ind}(0,a) = \int_0^{2\pi} \frac{a'(\theta)}{a(\theta)-z}\,\mathrm{d}\theta$$

si a est de classe C^1 ; si a est seulement continu, c'est l'intégrale

$$\mathrm{ind}(0,b) = \int_0^{2\pi} \frac{b'(\theta)}{b(\theta)-z}\,\mathrm{d}\theta$$

pour b de classe C^1 et suffisamment proche en norme uniforme de a. Enfin, on peut trouver une homotopie à valeur dans $\mathbb{C}\setminus\{0\}$ reliant a et e_m, c'est à dire qu'il existe une fonction $A(\theta,t)$ continue de $S^1\times[0,1]$ dans $\mathbb{C}\setminus\{0\}$ telle que $A(\cdot,0)=e_m$ et $A(\cdot,1)=a$.

Montrer que l'indice de T_a est l'opposé de l'indice de 0 par rapport à a. ▷

EXERCICE 5.31. ◁ Montrer que le spectre de T_a est la réunion de l'image de a et de l'ensemble des z dont l'indice par rapport à a n'est pas nul. ▷

Bibliographie

[1] Robert R. Bitmead and Brian D. O. Anderson. Asymptotically fast solution of Toeplitz and related systems of linear equations. *Linear Algebra Appl.*, 34 :103–116, 1980.

[2] David A. Cox, John Little, and Donal O'Shea. *Using algebraic geometry*, volume 185 of *Graduate Texts in Mathematics*. Springer, New York, second edition, 2005.

[3] 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.

[4] I. Gohberg, T. Kailath, and I. Koltracht. Efficient solution of linear systems of equations with recursive structure. *Linear Algebra Appl.*, 80 :81–113, 1986.

[5] 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.

[6] I. C. Gohberg and A. A. Semencul. The inversion of finite Toeplitz matrices and their continual analogues. *Mat. Issled.*, 7(2) :201–223, 290, 1972.

[7] G. Heinig and V. Olshevsky. The Schur algorithm for matrices with Hessenberg displacement structure. In *Structured matrices in mathematics, computer science, and engineering, II (Boulder, CO, 1999)*, volume 281 of *Contemp. Math.*, pages 3–15. Amer. Math. Soc., Providence, RI, 2001.

[8] 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.

[9] Thomas Kailath, Sun Yuan Kung, and Martin Morf. Displacement ranks of matrices and linear equations. *J. Math. Anal. Appl.*, 68(2) :395–407, 1979.

[10] Thomas Kailath and Ali H. Sayed. Displacement structure : theory and applications. *SIAM Rev.*, 37(3) :297–386, 1995.

[11] Martin Morf. *Fast algorithms for multivariable systems, Ph.D. Thesis*. Department of Electrical Engineering, Stanford University, Stanford, CA, 1974.

[12] V. Olshevsky and M. Amin Shokrollahi. A displacement approach to decoding algebraic codes. In *Fast algorithms for structured matrices : theory and applications (South Hadley, MA, 2001)*, volume 323 of *Contemp. Math.*, pages 265–292. Amer. Math. Soc., Providence, RI, 2003.

[13] 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.

[14] Victor Y. Pan. *Structured matrices and polynomials*. Birkhäuser Boston Inc., Boston, MA, 2001. Unified superfast algorithms.

[15] Jacobus Hendricus van Lint. *Introduction to coding theory*, volume 86 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, 1982. Problemy Matematicheskogo Analiza [Problems in Mathematical Analysis], 8.