

---

JOË SANGAR

# Déplacement et méthodes rapides de factorisation des matrices structurées

---

Mémoire d'initiation à la recherche

Sous la direction de Monsieur **Guillaume LEGENDRE**



Cycle **Pluridisciplinaire d'Études Supérieures**  
**Paris Sciences et Lettres**  
Troisième année (L3)

# Table des matières

<b>Remerciements</b>	<b>2</b>
<b>Introduction</b>	<b>3</b>
<b>1 Résolution classique des systèmes linéaires</b>	<b>4</b>
1.1 Factorisation LU . . . . .	6
1.2 Factorisation $LDM^T$ et factorisation de Cholesky . . . . .	10
1.3 Factorisation QR . . . . .	17
<b>2 Matrices et déplacement</b>	<b>20</b>
2.1 Matrices particulières . . . . .	20
2.2 Déplacement et rang de déplacement . . . . .	22
2.3 Propriétés de l'opérateur de déplacement . . . . .	28
<b>3 Exploitation de la structure de déplacement : méthodes de factorisation rapide</b>	<b>30</b>
3.1 Factorisation de rang plein et générateurs . . . . .	30
3.2 Algorithme GKO . . . . .	31
<b>A Boîte à outils</b>	<b>31</b>

## Remerciements

## Introduction

L'approximation et la résolution numérique de problèmes concrets conduit couramment à la résolution numérique de systèmes d'équations linéaires. Pour effectuer cette dernière, certaines méthodes qualifiées de directs, utilisent une procédé de factorisation pour ramener la résolution à celle d'un système équivalent, aisé à résoudre (triangulaire par exemple). La célèbre élimination de Gauss est l'une de ces méthodes. D'autres procédés, que l'on nomme itératifs, reposent quant à eux sur la construction d'une suite convergente vers la solution recherchée. On peut par exemple citer les méthodes de Gauss-Seidel ou de Jacobi par exemple. Lors de cas pratiques, les paramètres sont nombreux, les données relevées le sont donc d'autant plus. Les équations mettent alors en jeu des matrices de très grande dimension. La factorisation de matrices revêt donc une importance cruciale et le sujet suscite de ce fait beaucoup d'attention. Nous disposons aujourd'hui de techniques pour traiter ces problèmes de manière non naïve, nous y ferons allusion un peu plus loin. Malheureusement ces techniques sont assez lentes. Cependant, certains problèmes mènent à l'apparition de structures particulières dans les matrices qui les caractérisent. Nous allons donc voir qu'en utilisant judicieusement ces structures - grâce notamment à la notion de *déplacement* - nous serons en mesure d'obtenir des procédures de factorisation beaucoup plus rapide.

Dans ce travail, nous considérerons donc la résolution de systèmes linéaires dont la matrice possède une structure particulière qui peut être exploitée pour diminuer de manière importante le nombre d'opérations et l'espace mémoire requis pour la résolution, en exploitant le *déplacement*. Nous présenterons d'abord les méthodes de résolution classique des systèmes linéaires, puis nous introduirons le concept de déplacement et les matrices particulières qui y sont associées. Enfin nous étudierons les méthodes de factorisation rapides exploitant la structure de déplacement.

Il existe d'autres classes de matrices structurées que celles que nous allons introduire dans notre travail. Nous pouvons par exemple penser aux matrices bandes qui ne comportent des coefficients non nuls que sur leurs diagonales centrales. Nous examinerons également brièvement le cas des matrices symétriques (matrices égales à leur transposée).

Dans toute la suite,  $\mathbb{K}$  est un corps et pourra désigner aussi bien  $\mathbb{R}$  que  $\mathbb{C}$ . [KKM79]

# 1 Résolution classique des systèmes linéaires

Commençons par aborder les méthodes classiques pour résoudre les systèmes linéaires du type

$$Ax = b \tag{1}$$

Bien sûr cette équation n'admet une solution unique que si la matrice  $A$  est inversible. Un point important à prendre en compte et que dans la pratique, pour résoudre  $Ax = b$  où  $x$  est l'inconnue, on ne calcul quasiment jamais directement  $A^{-1}b$ . Le calcul de l'inverse est bien trop coûteux. Ces méthodes contournent donc ce problème par différents procédés. Parmi les méthodes classiques on peut distinguer deux grandes classes ; les méthodes directes et les méthodes itératives. Les méthodes directes visent à effectuer des transformations sur la matrice du système pour se ramener à un système équivalent plus simple à résoudre. Au sein de celles-ci on peut compter les méthodes de factorisation mais pas uniquement. La méthode du gradient conjuguée est par exemple aussi une méthode directe. Les méthodes itératives quant à elles reposent sur la construction d'une suite de vecteurs  $(x^{(k)})_{k \in \mathbb{N}}$  qui converge, on l'espère, vers la solution. Le principe est le suivant : A partir de l'initialisation  $x^{(0)} \in \mathbb{R}^n$ , on veut que la suite  $(x^{(k)})_{k \in \mathbb{N}}$  obtenue par une certaine relation de récurrence soit telle que,

$$\lim_{k \rightarrow +\infty} x^{(k)} = x = A^{-1}b$$

On oppose ses deux classes car dans le cas des méthodes directes le nombre d'opérations est en théorie fixé a priori alors que dans le cas des méthodes itératives il est en théorie infini. Dans la pratique le nombre d'opération des méthodes itératives est bien sûr fini. On utilise pour cela un test d'arrêt. Par exemple, étant donné une tolérance  $\epsilon$ , on arrêtera l'algorithme à l'étape  $k$  si la norme du résidu est inférieure à la tolérance. Avec la norme choisie, on a

$$\|r^{(k)}\| = \|b - Ax^{(k)}\|$$

Parmi les méthodes itératives, on compte par exemple les méthodes de Jacobi ou encore de Gauss-Seidel. Dans toute la suite, on se concentrera exclusivement sur les méthodes directes utilisant une factorisation.

**Définition 1.1.** *On entend par factorisation, l'écriture d'une matrice sous la forme d'un produit de matrices (deux ou plus) ayant des propriétés remarquables.*

*Soient  $A, B, C \in \mathbb{M}_n(\mathbb{K})$ , alors on a :*

$$A = BC$$

*Où les matrices  $B$  et  $C$  sont des matrices remarquables. Elles peuvent par exemple être diagonales, triangulaires ou orthogonales.*

Il existe différents types de factorisations. Ici, nous nous intéresserons principalement aux factorisations LU, QR et LDM<sup>T</sup>. L'intérêt de ces factorisations est qu'elles conduisent à un système linéaire associé simple à résoudre. On donne les solutions générales ainsi que les algorithmes de résolution. Par exemple, la solution d'un système linéaire triangulaire est obtenue en appliquant l'une des méthodes suivantes.

**Définition 1.2.** Soit  $A \in \mathbb{M}_n(\mathbb{K})$ ,  
Alors  $A$  est triangulaire supérieure (resp. inférieure) si  $\forall i, j$  tel que  $j < i$  ( resp.  $j > i$ )

$$a_{ij} = 0$$

Si  $A$  est une matrice triangulaire supérieure, et si aucun élément diagonal n'est nul, la solution du système (1) est :

$$\begin{cases} x_n = \frac{b_n}{a_{nn}} \\ x_i = \frac{b_i - \sum_{j=i+1}^n a_{ij}x_j}{a_{ii}} \end{cases} \quad \text{avec } i \in \llbracket n-1, 1 \rrbracket$$

On a alors l'algorithme :

---

**Algorithme 1** Algorithme de résolution d'un système linéaire triangulaire supérieure

---

**ENTRÉES:** le tableau contenant la matrice  $A$  et la liste contenant le vecteur  $b$ .

**SORTIES:** la liste contenant le vecteur solution  $x$ .

- 1:  $x(n) = \frac{b(n)}{A(n,n)}$
  - 2: **pour**  $i = n - 1$  à  $1$  **faire**
  - 3:    $\Sigma = 0$
  - 4:   **pour**  $j = i + 1$  à  $n$  **faire**
  - 5:      $\Sigma = \Sigma + A(i, j) * x(j)$
  - 6:   **finpour**
  - 7:    $x(i) = \frac{b(i) - \Sigma}{A(i,i)}$
  - 8: **finpour**
- 

Il s'agit donc d'un procédé de remontée.

De même, si  $A$  est une matrice triangulaire inférieure, et si aucun élément diagonal n'est nul, la solution du système (1) est :

$$\begin{cases} x_1 = \frac{b_1}{a_{11}} \\ x_i = \frac{b_i - \sum_{j=1}^{i-1} a_{ij}x_j}{a_{ii}} \end{cases} \quad \text{avec } i \in \llbracket 2, n \rrbracket$$

On a alors l'algorithme :

---

**Algorithme 2** Algorithme de résolution d'un système linéaire triangulaire inférieure

---

**ENTRÉES:** le tableau contenant la matrice  $A$  et la liste contenant le vecteur  $b$ .

**SORTIES:** la liste contenant le vecteur solution  $x$ .

```
1:  $x(1) = \frac{b(1)}{A(1,1)}$ 
2: pour  $i = 2$  à  $n$  faire
3:    $\Sigma = 0$ 
4:   pour  $j = 1$  à  $i - 1$  faire
5:      $\Sigma = \Sigma + A(i, j) * x(j)$ 
6:   finpour
7:    $x(i) = \frac{b(i) - \Sigma}{A(i, i)}$ 
8: finpour
```

---

Il s'agit donc d'un procédé de descente.

### 1.1 Factorisation LU

Avant de nous pencher sur la factorisation LU, faisons un petit rappel sur l'élimination de Gauss. Cette dernière a pour but de transformer le système (1) en un système de la forme

$$Ux = \widehat{b} \quad (2)$$

Où  $U$  est une matrice triangulaire supérieure et  $\widehat{b}$  est la modification du vecteur  $b$  en conséquence. On peut alors aisément résoudre (2) par un procédé de remontée. L'élimination de Gauss repose sur le fait que l'on obtient  $U$  à partir de  $A$  en ne réalisant que des opérations qui préservent les solutions du système, (des combinaisons linéaires d'équations entre elles) c'est à dire que les deux systèmes sont équivalents, (1) et (2) ont donc bien la même solution.

**Définition 1.3.** Soit  $A \in \mathbb{M}_n(\mathbb{K})$

Alors la factorisation (ou décomposition) LU de la matrice  $A$  revient à l'écrire sous la forme :

$$A = LU$$

Où  $L$  est une matrice triangulaire inférieure (Lower) et  $U$  une matrice triangulaire supérieure (Upper).

Le principe de la factorisation LU consiste donc à transformer le système (1) en un système équivalent :

$$LUx = b$$

Puis on résout

$$Ly = b$$

L'obtention de  $y$  se fait de manière aisée par méthode de la descente car  $L$  est triangulaire inférieure. On résout enfin

$$Ux = y$$

Là encore la résolution est aisée car le système est triangulaire et on obtient  $x$ .

**Remarque 1.** La factorisation  $LU$  est intéressante car, comme on le verra plus loin, les matrices  $L$  et  $U$  ne dépendent pas du second membre  $b$  de l'équation (1), mais uniquement de la matrice  $A$ . La factorisation n'a donc pas besoin d'être recalculée pour des systèmes qui ont la même matrice  $A$ , mais dont le second membre diffère. Ce faisant, on gagne à la fois en effort de calcul et en effort de stockage car la factorisation est stockée là où se trouve  $A$ .

Il est alors intéressant de constater que la décomposition  $LU$  est équivalente à l'élimination de Gauss. S'en est en fait une *traduction* matricielle. Toutes les transformations que l'on effectue sur la matrice  $A$  lors de l'élimination de Gauss peuvent être interprétées sous forme matricielle. Si l'on construit la matrice qui rassemble toutes ces transformations successives, alors la matrice  $L$  est l'inverse de cette dernière. La matrice  $U$  est obtenue à l'issue de l'élimination des coefficients sous-diagonaux de  $A$ .

**Définition 1.4.** Soit  $A \in \mathbb{M}_n(\mathbb{K})$

Une sous-matrice de  $A$  est une matrice obtenue en sélectionnant une partie  $I \subset \{1, \dots, n\}$  de ses lignes et une partie  $J \subset \{1, \dots, n\}$ . Puis en constituant par concaténation la matrice des coefficients qui se trouvent dans l'intersection des lignes et des colonnes choisies.

**Définition 1.5.** En restant dans le cadre de la définition 1.4. Une sous-matrice est dite principale lorsque  $I = J$ , de plus l'ensemble doit avoir la forme  $I = \llbracket 1, k \rrbracket$  pour  $1 \leq k \leq n$

**Définition 1.6.** En restant dans le cadre de la définition 1.4. Un mineur principal d'une matrice  $A$  est le déterminant d'une des sous-matrices principales de  $A$ .

**Définition 1.7.** En restant dans le cadre de la définition 1.4. L'ordre d'une sous-matrice principale correspond au cardinal de  $I$ .

**Définition 1.8.** En restant dans le cadre de la définition 1.4. Une sous matrice principale  $A_i$  est une sous matrice telle que  $I = \llbracket 1, i \rrbracket$  pour  $1 \leq i \leq n$ .

**Théorème 1. Factorisation  $LU$**

Soit  $A \in \mathbb{M}_n(\mathbb{K})$ , la factorisation  $LU$  de  $A$  telle que  $l_{ii} = 1 \quad \forall i \in \llbracket 1, n \rrbracket$  existe et est unique si et seulement si les sous-matrices principales  $A_i$  de  $A$  d'ordre  $i \in \llbracket 1, n-1 \rrbracket$  sont inversibles.

*Démonstration.* Nous allons effectuer un raisonnement par double implication.

Supposons que les sous-matrices principales  $A_i$  de  $A$  sont inversibles  $\forall i \in \llbracket 1, n-1 \rrbracket$ .

Nous allons montrer par récurrence sur  $i$  l'existence et l'unicité de la factorisation  $LU$  de  $A$ . Notre propriété est la suivante.

$P_{i-1}$  : il existe une unique factorisation  $LU$  de  $A_{i-1}$  telle que

$$A_{i-1} = L^{(i-1)}U^{(i-1)} \text{ et } \forall k \in \llbracket 1, i-1 \rrbracket, \quad l_{kk}^{(i-1)} = 1$$

Pour  $i = 1$  la propriété est trivialement vraie, donc  $P_0$  est vraie.

Soit  $i \in \mathbb{N}^*$  tel que  $P_{i-1}$  soit vraie, tentons de montrer que  $P_i$  est vraie.

On décompose  $A_i$  en bloc comme suit :

$$A_i = \begin{pmatrix} A_{i-1} & \mathbf{c} \\ \mathbf{d}^T & a_{ii} \end{pmatrix} \quad (3)$$

où  $\mathbf{c}$  et  $\mathbf{d}$  sont des vecteurs de  $\mathbb{K}^{i-1}$ . On peut alors chercher la factorisation LU de  $A_i$ . On sait que  $L^{(i)}$  est triangulaire inférieure et que ses coefficients diagonaux sont tous égaux à 1. On sait également que  $U^{(i)}$  triangulaire supérieure. On en déduit l'équation suivante :

$$A_i = \begin{pmatrix} A_{i-1} & \mathbf{c} \\ \mathbf{d}^T & a_{ii} \end{pmatrix} = L^{(i)}U^{(i)} = \begin{pmatrix} E & \mathbf{0} \\ \mathbf{f}^T & 1 \end{pmatrix} \begin{pmatrix} G & \mathbf{h} \\ \mathbf{0}^T & u_{ii} \end{pmatrix}$$

où  $E, G \in \mathbb{M}_i(\mathbb{K})$ ,  $f, h \in \mathbb{K}^{i-1}$  et  $u_{ii} \in \mathbb{K}$ . En effectuant le produit matriciel et en procédant par identification on obtient les équations :

$$\begin{aligned} A_{i-1} &= EG \\ Eh &= c \\ f^T G &= d^T \\ f^T h + u_{ii} &= a_{ii} \end{aligned}$$

On en déduit donc :

$$\begin{aligned} E &= L^{(i-1)} \\ G &= U^{(i-1)} \end{aligned}$$

Ainsi :

$$A_i = \begin{pmatrix} L^{(i-1)} & \mathbf{0} \\ \mathbf{f}^T & 1 \end{pmatrix} \begin{pmatrix} U^{(i-1)} & \mathbf{h} \\ \mathbf{0}^T & u_{ii} \end{pmatrix} \quad (4)$$

De plus on a les équations :

$$\begin{aligned} L^{(i-1)}h &= c \\ f^T U^{(i-1)} &= d^T \end{aligned}$$

Sachant que  $A^{(i-1)}$  est inversible on a  $\det(A^{(i-1)}) = \det(L^{(i-1)}) \times \det(U^{(i-1)}) \neq 0$  donc  $\det(A^{(i-1)}) \neq 0$  et  $\det(L^{(i-1)}) \neq 0$  donc  $L^{(i-1)}$  et  $U^{(i-1)}$  sont inversibles. De ce fait les vecteurs  $h$  et  $f$  existent et sont uniques. Le vecteur  $u_{ii}$  est également unique. On

a donc démontré par récurrence que si les sous-matrices principales  $A_i$  de  $A$  d'ordre  $i \in \llbracket 1, n-1 \rrbracket$  sont inversibles, alors il existe une unique factorisation LU de  $A$  telle que pour  $i \in \llbracket 1, n-1 \rrbracket$ ,  $l_{ii} = 1$ .

Supposons qu'il existe une unique factorisation LU de  $A$  telle que pour  $i \in \llbracket 1, n-1 \rrbracket$ ,  $l_{ii} = 1$ . Il est alors nécessaire d'effectuer un disjonction de cas selon que  $A$  est inversible ou non.

Soit  $A$  est inversible,

on a alors, pour  $i \in \llbracket 1, n-1 \rrbracket$ ,  $A_i = L^{(i)}U^{(i)}$ , donc

$$\det(A_i) = \det(L^{(i)}) \times \det(U^{(i)}) = \left( \prod_{k=1}^i l_{kk} \right) \times \det(U^{(i)}) = \det(U^{(i)}) = \prod_{k=1}^i u_{kk} \quad (5)$$

Pour  $i = n$  on a  $A_n = A$ , or  $A$  est inversible donc  $\det(A) = \prod_{k=1}^n u_{kk} \neq 0$ , on en déduit donc que  $\forall i \in \llbracket 1, n-1 \rrbracket$ ,  $\det(A_i) \neq 0$ . Ainsi,  $\forall i \in \llbracket 1, n-1 \rrbracket$  les sous-matrices principales  $A_i$  de  $A$  sont inversibles.

Soit  $A$  n'est pas inversible,

Supposons qu'au moins un des termes diagonaux de  $U$  soit nul. Notons  $u_{pp}$  le terme nul de  $U$  dont l'indice est le plus petit. Comme on l'a vu avec l'équation (4) on peut effectuer la factorisation jusqu'à la  $k+1$ -ème étape. A partir de là, la matrice  $U^{(k)}$  n'est plus inversible donc on a plus unicité du vecteur  $f$ . On perd également l'unicité de la factorisation. Pour éviter cela avant la factorisation totale de la matrice  $A$ , il faut que les  $u_{ii}$  soient non nul jusqu'au  $n-1$ -ème terme. Ainsi, d'après l'égalité (5), toutes les sous-matrices principales  $A_k$  doivent être inversibles pour  $k \in \llbracket 1, n-1 \rrbracket$ . □

Plus généralement, même si  $A$  n'est pas inversible, on peut toujours trouver une matrice de permutation  $P$  telle que que  $PA$  soit factorisable LU. Par commodité, on pourra nommer cette factorisation factorisation  $PLU$ . On en déduit que toute matrice  $A$  non inversible est factorisable en ce sens. Dans cette factorisation  $L$  et  $U$  restent les mêmes matrices triangulaires inférieure et supérieure. Cette factorisation est moins contraignante que la factorisation LU.

Nous pourrions rajouter que pour des raisons de stabilité numériques importante cette permutation est faite selon une stratégie de choix de pivot. En effet, si on procède uniquement à des échanges de lignes - ce qu'on appelle un procédé de *pivot partiel* - lors de la procédure, alors avec les matrices de permutation successives on parvient a

$$PA = LU$$

Si on procède à la fois à des échanges de lignes et de colonnes - ce qu'on appelle un procédé de *pivot total* - alors une seconde matrice de permutation doit être prise en compte. On obtient alors

$$PAQ = LU$$

Bien sûr, ces échanges de lignes et de colonnes (sauf dans des cas très particuliers) ne permettent pas de conserver la structure de la matrice si cette dernière en possédait une.

L'algorithme que nous donnons ci-dessous est écrit en pseudo-code. Il s'agit d'une première version de l'algorithme dit orienté colonnes (du fait de l'ordre dans lequel on effectue les boucles). Il existe également des algorithmes orientés lignes.

---

**Algorithme 3** Algorithme de factorisation LU

---

**ENTRÉES:** le tableau contenant la matrice  $A$ .

**SORTIES:** la partie triangulaire inférieure stricte de la matrice  $L$  et la partie triangulaire supérieure de la matrice  $U$ , stockées dans le tableau contenant la matrice  $A$  en entrée.

```

1: pour  $k = 1$  à  $n - 1$  faire
2:   si  $A(k, k) \neq 0$  alors
3:     pour  $i = k + 1$  à  $n$  faire
4:        $A(i, k) = A(i, k) / A(k, k)$ 
5:     finpour
6:     pour  $j = k + 1$  à  $n$  faire
7:       pour  $i = k + 1$  à  $n$  faire
8:          $A(i, j) = A(i, j) - A(i, k) * A(k, j)$ 
9:       finpour
10:    finpour
11:   sinon
12:     arrêt
13:   fin
14: finpour

```

---

Nous pouvons dès lors nous atteler à calculer le coup de cet algorithme.

Nous constatons un coût en  $\mathcal{O}(\frac{2}{3}n^3)$ . Cet algorithme peut être implémenté de différentes manières selon la méthode de pivot que l'on choisit. En effet, on a par exemple le choix entre un pivot total ou un pivot partiel. Le choix de la méthode de pivot peut avoir une grande importance en fonction du calculateur que l'on utilise comme stipulé dans l'article [GPS90].

## 1.2 Factorisation $LDM^T$ et factorisation de Cholesky

La factorisation  $LDM^T$  est un autre type de factorisation que l'on tire des outils de la factorisation LU.

**Définition 1.9.** Soit  $A \in \mathbb{M}_n(\mathbb{K})$

Alors la factorisation  $LDM^T$  de la matrice revient à l'écrire sous la forme.

$$A = LDM^T$$

Où  $L$  est une matrice triangulaire inférieure,  $D$  est une matrice diagonale et  $M^T$  une matrice triangulaire supérieure.

Le principe de la factorisation  $LDM^T$  consiste donc à transformer le système (1) en un système équivalent :

$$LDM^T x = b$$

Puis on résout

$$Ly = b$$

L'obtention de  $y$  se fait de manière aisée car  $L$  est triangulaire. On résout ensuite

$$Dz = y$$

$D$  étant diagonale  $z$  s'obtient par un calcul trivial. On résout enfin

$$M^T x = z$$

Là encore la résolution est aisée car le système est triangulaire et on obtient  $x$  par remontée.

**Définition 1.10.** On appelle symbole de Kronecker la fonction telle que :

$$\delta_{ij} = \begin{cases} 1 & \text{si } i = j \\ 0 & \text{si } i \neq j \end{cases}, \text{ on a aussi } \delta_{i \leq j} = \begin{cases} 1 & \text{si } i \leq j \\ 0 & \text{si } i > j \end{cases}$$

**Théorème 2. Factorisation  $LDM^T$**

Soit  $A \in \mathbb{M}_n(\mathbb{K})$ , si tout les mineurs principaux de  $A$  sont non nuls, alors la factorisation  $LDM^T$  de  $A$  telle que  $m_{ii} = l_{ii} = 1$ ,  $\forall i \in \llbracket 1, n \rrbracket$  est unique.

**Remarque 2.** Il s'agit en fait d'un corollaire du Théorème 1. de factorisation  $LU$  comme nous allons à présent le voir dans la démonstration.

*Démonstration.* Tout les mineurs principaux de  $A$  sont non nuls donc en particuliers tout les déterminants des sous-matrices principales  $A_i$  sont non nuls. Donc toutes les sous-matrices  $A_i$  sont inversibles. D'après le Théorème 1, on peut donc affirmer que  $A$  admet une factorisation  $LU$  unique telle que  $\forall i \in \llbracket 1, n \rrbracket$   $l_{ii} = 1$ .

Si on se donne  $D \in \mathbb{M}_n(\mathbb{K})$  telle que :

$$D = \text{diag}(u_{11}, \dots, u_{nn})$$

Comme  $U$  est inversible on a que  $\forall i \in \llbracket 1, n \rrbracket$ ,  $u_{ii} \neq 0$ . On en déduit donc que

$$D^{-1} = \text{diag}\left(\frac{1}{u_{11}}, \dots, \frac{1}{u_{nn}}\right)$$

On peut alors écrire

$$A = LU = LDD^{-1}U$$

Si on pose

$$M^T = D^{-1}U$$

$$M_{ij}^T = \sum_{k=1}^n D_{ik}^{-1} U_{kj} \delta_{k=i} \delta_{k \leq j}$$

Or, on a que  $\delta_{k=i} \delta_{k \leq j} \implies \delta_{i \leq j}$ , donc la matrice est triangulaire supérieure

$$\begin{aligned} &= \sum_{k=1}^j D_{ik}^{-1} U_{kj} \delta_{k=i} \\ &= \sum_{k=1}^j \frac{1}{U_{ik}} \times U_{kj} \times \delta_{k=i} \\ &= \frac{U_{ij}}{U_{ii}} \end{aligned}$$

Donc

$$M_{ii}^T = \sum_{k=1}^i \frac{1}{U_{ik}} \times U_{ki} \times \delta_{k=i} = \frac{U_{ii}}{U_{ii}} = 1$$

Ainsi  $M^T$  est bien une matrice triangulaire supérieure telle que  $\forall i \in \llbracket 1, n \rrbracket, m_{ii} = 1$ .

Ainsi on a bien

$$A = LDM^T$$

□

Comme le montre cette démonstration, on peut extraire la factorisation LDM<sup>T</sup> directement de la factorisation LU. On peut alors se contenter de modifier légèrement l'algorithme de factorisation LU pour obtenir l'algorithme de factorisation LDM<sup>T</sup>. On obtient alors l'algorithme suivant :

---

**Algorithme 4** Algorithme de factorisation LDM<sup>T</sup>

---

**ENTRÉES:** le tableau contenant la matrice  $A$ .

**SORTIES:** la partie triangulaire inférieure stricte de la matrice  $L$  et la partie triangulaire supérieure stricte de la matrice  $M^T$  ainsi que la matrice  $D$  stockées dans le tableau contenant la matrice  $A$  en entrée.

```
1: pour  $k = 1$  à  $n - 1$  faire
2:   si  $A(k, k) \neq 0$  alors
3:     pour  $i = k + 1$  à  $n$  faire
4:        $A(i, k) = A(i, k) / A(k, k)$ 
5:     finpour
6:     pour  $j = k + 1$  à  $n$  faire
7:       pour  $i = k + 1$  à  $n$  faire
8:          $A(i, j) = A(i, j) - A(i, k) * A(k, j)$ 
9:       finpour
10:    finpour
11:    pour  $i = 1$  à  $n - 1$  faire
12:      pour  $j = i + 1$  à  $n$  faire
13:         $A(i, j) = \frac{1}{A(i, i)} * A(i, j)$ 
14:      finpour
15:    finpour
16:  sinon
17:    arrêt
18:  finsi
19: finpour
```

---

Les boucles que l'on rajoute ont un coût de  $\mathcal{O}(n^2)$ , l'algorithme de factorisation LDM<sup>T</sup> a donc le même coup asymptotique que l'algorithme de factorisation LU.

Lorsque la matrice  $A$  est symétrique on peut dégager une propriété intéressante.

**Proposition 1.** *Soit  $A \in \mathbb{M}_n(\mathbb{K})$ , si  $A$  est symétrique et admet une factorisation LDM<sup>T</sup>, alors on a  $L = M$ , c'est à dire*

$$A = LDL^T$$

*Démonstration.* □

Cela permet de gagner en effort de calcul et de stockage par rapport à la factorisation LU. De cette observation, on peut déduire une nouvelle factorisation qui s'applique uniquement aux matrices symétriques définies positives, il s'agit de la factorisation de Cholesky.

**Définition 1.11.** *Soit  $A \in \mathbb{M}_n(\mathbb{R})$*

*Alors  $A$  est définie positive si pour tout vecteurs  $x$  non nuls de  $\mathbb{R}^n$  on a :*

$$x^T A x > 0$$

**Définition 1.12.** On nomme l'ensemble des matrices carrées symétriques définies positives de  $\mathbb{K}$  de dimensions  $n$  :  $\mathbb{S}_n^+(\mathbb{K})$

**Définition 1.13.** Soit  $A \in \mathbb{S}_n^+(\mathbb{K})$

Alors la factorisation de Cholesky de la matrice  $A$  revient à l'écrire sous la forme :

$$A = BB^T$$

Où  $B$  est une matrice triangulaire inférieure.

Le principe de la factorisation de Cholesky consiste donc à transformer le système (1) en un système équivalent :

$$BB^T x = b$$

Puis on résout :

$$By = b$$

L'obtention de  $y$  se fait de manière aisée car  $B$  est triangulaire. On résout enfin

$$B^T x = y$$

Là encore la résolution se fait de manière aisée car le système est triangulaire et on obtient  $x$ .

**Proposition 2. Critère de Sylvester**

Soit  $A \in \mathbb{M}_n(\mathbb{R})$ ,  $A$  est définie positive si et seulement si tout ces mineurs principaux sont strictement positifs.

*Démonstration.* ?

□

**Théorème 3. Factorisation de Cholesky**

Soit  $A \in \mathbb{S}_n^+(\mathbb{R})$  alors il existe une unique matrice triangulaire inférieure  $B$ , dont tout les éléments diagonaux sont strictement positifs, telle que :

$$A = BB^T$$

*Démonstration.* Nous allons commencer par prouver l'unicité.

Supposons qu'il existe deux matrices triangulaires inférieures  $B_1, B_2 \in \mathbb{M}_n(\mathbb{R})$  telles que :

$$A = B_1 B_1^T = B_2 B_2^T \tag{6}$$

Les éléments diagonaux de  $B_1$  et  $B_2$  sont tous strictement positifs donc  $\det(B_1) \neq 0$  et  $\det(B_2) \neq 0$  donc les deux matrices sont inversibles. On peut donc déduire l'égalité

$$B_2^{-1} B_1 = B_2^T (B_1^T)^{-1}$$

D'après la Proposition 1.,  $B_2^{-1}$  est une matrice triangulaire inférieure et  $(B_1^T)^{-1}$  est une matrice triangulaire supérieure. On sait également que  $B_1$  est une matrice triangulaire

inférieure et  $B_2^T$  une matrice triangulaire supérieure. On peut donc en déduire que  $B_2^{-1}B_1$  est une matrice triangulaire inférieure et que  $B_2^T(B_1^T)^{-1}$  est une matrice triangulaire supérieure. Or il y a égalité entre les deux matrices, elles sont donc toutes les deux à fois triangulaires supérieures et triangulaires inférieures. Ce sont donc des matrices diagonales. Ainsi, il existe une matrice diagonale  $D$  telle que :

$$D = B_2^{-1}B_1 \tag{7}$$

Par conséquent

$$B_1 = B_2D$$

D'après (6), on a alors

$$B_2B_2^T = B_1B_1^T = (B_2D)(B_2D)^T = B_2DD^TB_2^T$$

En identifiant on déduit

$$DD^T = D^2 = I_n$$

Ce qui équivaut à,  $\forall i \in \llbracket 1, n \rrbracket$   $d_{ii}^2 = 1$ , sachant que  $\forall i \in \llbracket 1, n \rrbracket$   $d_{ii} > 0$ , on en déduit  $d_{ii} = 1$  donc  $D = I_n$  donc d'après (7)

$$B_1 = (B_2^{-1})^{-1} = B_2$$

Attelons désormais à la preuve de l'existence.

On a  $A \in \mathbb{S}_n^+(\mathbb{K})$ , par la Proposition 3., on sait que les mineurs des sous-matrices principales  $A_i$  pour  $1 \leq i \leq n$  sont strictement positives, on peut donc avancer que d'après le Théorème 1., la matrice  $A$  admet une unique factorisation LU. De plus, les coefficients diagonaux de  $U$  sont strictement positifs. En effet, on a :

$$\forall i \in \llbracket 1, n \rrbracket, \prod_{k=1}^i u_{kk} = \det(A_i) > 0$$

On peut donc se donner  $D \in \mathbb{M}_n(\mathbb{K})$  tel que

$$D = \text{diag}(\sqrt{u_{11}}, \dots, \sqrt{u_{nn}}) \text{ donc } D^{-1} = \text{diag}\left(\frac{1}{\sqrt{u_{11}}}, \dots, \frac{1}{\sqrt{u_{nn}}}\right)$$

On a alors

$$A = LDD^{-1}U$$

On pose

$$B = LD \text{ et } C = D^{-1}U \text{ qui donnent } A = BC$$

Comme  $A$  est symétrique

$$A = A^T = (BC)^T = C^TB^T = BC \text{ donc } C^TB^T = BC$$

$B$  est une matrice triangulaire inférieure dont tout les éléments diagonaux sont strictement positifs donc  $\det(B) \neq 0$  donc  $B$  est inversible.

$C$  est une matrice triangulaire supérieure dont tout les éléments diagonaux sont strictement positifs donc  $\det(C) \neq 0$  donc  $C$  est inversible. On en déduit

$$C(B^T)^{-1} = B^{-1}C^T$$

$C(B^T)^{-1}$  est triangulaire supérieure et  $B^{-1}C^T$  est triangulaire inférieure. Les deux matrices sont donc à la fois triangulaires inférieures et triangulaires supérieures. Elles sont donc diagonales. De plus leur coefficients diagonaux sont égaux à un. On à donc

$$C(B^T)^{-1} = B^{-1}C^T = I_n \text{ donc } C = ((B^T)^{-1})^{-1} = B^T$$

□

---

**Algorithme 5** Algorithme de factorisation de Cholesky

---

**ENTRÉES:** le tableau contenant la matrice  $A$ .

**SORTIES:** le tableau contenant la matrice  $B$

```

1: pour  $k = 1$  à  $n - 1$  faire
2:   si  $A(k, k) \leq 0$  alors
3:     Erreur : Pivot nul ou négatif
4:   Arrêt
5:   sinon
6:      $A(k, k) = \text{sqrt}(A(k, k))$ 
7:     pour  $l = k + 1$  à  $n$  faire
8:        $A(l, k) = A(l, k) / A(k, k)$ 
9:     finpour
10:    pour  $j = k + 1$  à  $n$  faire
11:      pour  $m = j$  à  $n$  faire
12:         $A(m, j) = A(m, j) - A(m, k) * A(j, k)$ 
13:      finpour
14:    finpour
15:  finsi
16: finpour
17:  $A(n, n) = \text{sqrt}(A(n, n))$ 

```

---

Pour cet algorithme on a un coup de  $\mathcal{O}(\frac{n^3}{3})$  en effet on a  $\sum_{k=2}^{n-1} [(k-1) + \sum_{i=k+1}^n 1] + 2n - 2 \approx \frac{n^3}{6}$  multiplication et divisions ainsi que  $\frac{n^3-n}{6} \approx \frac{n^3}{6}$  additions et soustractions. On constate donc une diminution de moitié du coup par rapport à la factorisation LU. La symétrie qui est en soit une forme de structure permet d'ores et déjà d'obtenir une factorisation plus rapide.

**Remarque 3.** Dans le calcul de la complexité nous avons omis le coup de l'extraction des racines. Dans la pratiques ce dernier est bien plus couteux qu'une multiplication ou division. Cependant, ici on peut se permettre de ne pas les prendre en compte car il n'y en a que  $n$ .

### 1.3 Factorisation QR

Contrairement à la factorisation LU, la factorisation QR peut être utilisée pour la résolution de systèmes surdéterminés, c'est à dire avec des matrices non carrées. On l'utilise également pour le calcul des valeurs propres et pour la résolution des moindres carrés.

**Définition 1.14.** Soit  $A \in \mathbb{M}_n(\mathbb{K})$

Alors la factorisation QR de la matrice  $A$  revient à l'écrire sous la forme :

$$A = QR$$

Où  $Q$  est une matrice orthogonale (ou unitaire si on se trouve dans  $\mathbb{C}$ ) et  $R$  une matrice triangulaire supérieure.

**Définition 1.15.** Soit  $Q \in \mathbb{M}_n(\mathbb{R})$

Alors  $Q$  est orthogonale si

$$QQ^T = Q^T Q = I$$

On en déduit donc que

$$Q^{-1} = Q^T$$

**Définition 1.16.** Soit  $Q \in \mathbb{M}_n(\mathbb{C})$

Alors  $Q$  est unitaire si

$$QQ^* = Q^* Q = I$$

On rappelle que  $Q^* = \overline{Q^T} = \overline{Q}^T$

Le principe de la factorisation QR consiste à transformer le système (1) en un système équivalent :

$$QRx = b$$

Puis on résout

$$Qy = b$$

L'obtention de  $y$  se fait de manière aisée. Il s'agit d'une des rares occurrences où l'on passe par le calcul de l'inverse. Ici,  $Q$  est orthogonale donc  $Q^{-1} = Q^T$ , ce qui signifie que l'inverse est très facile à obtenir. On a donc

$$y = Q^{-1}b = Q^T b$$

On résout enfin

$$Rx = y$$

Là encore la résolution est aisée car le système est triangulaire et on obtient  $x$ .

**Remarque 4.** La factorisation QR est intéressante car les matrices orthogonales disposent de bonnes propriétés. Grâce à elles, les matrices  $A$  et  $R$  ont le même conditionnement et le même déterminant.

**Théorème 4.** Soit  $A \in \mathbb{M}_n(\mathbb{K})$ ,

Si  $A$  est inversible alors la factorisation  $QR$  de  $A$  telle que les éléments diagonaux de  $R$  soient strictement positifs existe et est unique.

*Démonstration.* Nous allons commencer par prouver l'unicité.

Supposons qu'il existe deux matrices orthogonales  $Q_1, Q_2 \in \mathbb{M}_n(\mathbb{R})$  ainsi que deux matrices triangulaires supérieure  $R_1, R_2 \in \mathbb{M}_n(\mathbb{R})$  telles que :

$$A = Q_1 R_1 = Q_2 R_2 \quad (8)$$

Comme  $A$  est inversible on a  $\det(A) \neq 0$  donc  $\det(Q_1 R_1) = \det(Q_2 R_2) \neq 0$  donc  $\det(Q_1), \det(Q_2), \det(R_1), \det(R_2)$ , sont tous non nuls donc les matrices sont toutes inversibles. On peut alors déduire de (8) que

$$Q_2^T Q_1 = R_2 R_1^{-1}$$

Si on pose

$$T = R_2 R_1^{-1}$$

On a alors

$$\begin{aligned} T^T T &= (R_2 R_1^{-1})^T (R_2 R_1^{-1}) \\ &= (Q_2^T Q_1)^T (Q_2^T Q_1) \\ &= Q_1^T Q_2 Q_2^T Q_1 \\ &= Q_1^T I_n Q_1 \\ &= I_n \end{aligned}$$

On obtient donc l'égalité  $T^T T = I_n$  dans laquelle on reconnaît une factorisation de Cholesky de  $I_n$ . D'après le Théorème 3. on sait que cette factorisation est unique. Or on a que  $I_n = I_n I_n^T$ , on en déduit donc que

$$T = I_n$$

C'est à dire

$$R_2 R_1^{-1} = I_n \quad \text{et} \quad Q_2^T Q_1 = I_n$$

Donc

$$R_2 = (R_1^{-1})^{-1} = R_1 \quad \text{et} \quad Q_1 = (Q_2^T)^{-1} = Q_2$$

Ainsi, si la factorisation QR existe, elle est bel et bien unique.

Prouvons maintenant son existence. La matrice  $A$  étant inversible, ses colonnes, notées  $\mathbf{a}_1, \dots, \mathbf{a}_n$  forment une base de  $\mathbb{R}^n$ . On peut alors obtenir une base orthonormée  $(q_1, \dots, q_n)$  de  $\mathbb{R}^n$  à partir de la famille  $\mathbf{a}_1, \dots, \mathbf{a}_n$ , en appliquant le procédé de Gram-Schmidt, c'est à dire en procédant de la manière suivante :

$$q_1 = \frac{\mathbf{a}_1}{\|\mathbf{a}_1\|_2}$$

$$\forall j \in \llbracket 1, n-1 \rrbracket \quad \tilde{q}_{j+1} = \mathbf{a}_{j+1} - \sum_{k=1}^j (q_k, \mathbf{a}_{j+1}) q_k, \quad q_{j+1} = \frac{\tilde{q}_{j+1}}{\|\tilde{q}_{j+1}\|_2}$$

On en déduit alors que

$$\mathbf{a}_j = \sum_{i=1}^j r_{ij} q_i,$$

où on a

$$\begin{cases} r_{jj} = \|\mathbf{a}_j - \sum_{k=1}^{j-1} (q_k, \mathbf{a}_j) q_k\|_2 \\ r_{ij} = (q_i, \mathbf{a}_j) & \text{pour } i \in \llbracket 1, j-1 \rrbracket \\ r_{ij} = 0 & \text{pour } j \in \llbracket 1, n \rrbracket \text{ et } i \in \llbracket j+1, n \rrbracket \end{cases}$$

Si on note  $R$  la matrice triangulaire supérieure constituée des coefficients  $r_{ij}$  et  $Q$  la matrice orthogonale dont les colonnes sont les vecteurs  $q_j$  alors on a bien  $A = QR$ .  $\square$

Nous donnons ci-après l'algorithme qui nous permet d'obtenir la factorisation QR par le procédé d'orthonormalisation de Gram-Schmidt.

---

**Algorithme 6** Algorithme de factorisation QR d'une matrice inversible par le procédé d'orthonormalisation de Gram-Schmidt

---

**ENTRÉES:** le tableau contenant la matrice  $A$ .

**SORTIES:** les tableaux contenant la matrice orthogonale  $Q$  et la matrice triangulaire supérieure  $R$ .

- 1: **pour**  $j = 1$  à  $n$  **faire**
  - 2:    $Q(1 : n, j) = A(1 : n, j)$
  - 3:   **pour**  $i = 1$  à  $j - 1$  **faire**
  - 4:      $R(i, j) = Q(1 : n, i)^T * Q(1 : n, j)$
  - 5:      $Q(1 : n, j) = Q(1 : n, j) - R(i, j) * Q(1 : n, i)$
  - 6:   **finpour**
  - 7:    $R(j, j) = \|Q(1 : n, j)\|_2$
  - 8:    $Q(1 : n, j) = Q(1 : n, j) / R(j, j)$
  - 9: **finpour**
- 

On constate que l'algorithme à un coût de  $\mathcal{O}(\frac{4n^3}{3})$ . Il convient de mentionner que cet algorithme n'est pas celui utilisé dans la pratique pour des raisons de stabilité numérique. En effet, l'orthogonalité est une propriété difficile à conserver par le procédé de Gram-Schmidt car elle produit des valeurs très petites, de l'ordre de la précision machine. A cause de cela, on peut obtenir en sortie une matrice non orthogonale. On utilise donc, dans les faits, une version équivalente mais modifiée de cet algorithme, qui lui aussi à un coût en  $\mathcal{O}(\frac{4n^3}{3})$ .

## 2 Matrices et déplacement

### 2.1 Matrices particulières

**Définition 2.1.** Soit  $A \in \mathbb{M}_n(\mathbb{K})$ ,  $A$  est dite creuse si elle n'a que  $\mathcal{O}(n)$  coefficients non nuls.

**Définition 2.2.** Soit  $A \in \mathbb{M}_n(\mathbb{K})$ ,  $A$  est dite structurée si elle n'est pas creuse et que ces  $n^2$  coefficients ne dépendent que de  $\mathcal{O}(n)$  entrées.

**Remarque 5.** De plus les coefficients doivent être distribués de manière à ce que l'on puisse travailler "simplement" sur la matrice. On voit ici affleurer la notion de déplacement sur laquelle on se penchera plus en détails dans la partie suivante.

Il existe plusieurs types de matrices structurées. Les plus classiques et celles auxquelles nous allons nous intéresser, sont les matrices de Cauchy, Hankel, Toeplitz et Vandermonde. Nous allons les introduire et tenter de donner quelques exemples d'applications.

- (a) Une matrice de Cauchy (C) est une matrice carrée que l'on peut définir par deux vecteurs :

$$X = (x_1 \quad \dots \quad x_n)^T \quad \text{et} \quad Y = (y_1 \quad \dots \quad y_n)^T \quad \text{tel que} \quad \forall (i, j), x_i \neq y_j$$

Les coefficients  $C_{ij}$  de la matrice vérifient la relation :

$$C_{ij} = \frac{1}{x_i - y_j}$$

On obtient donc une matrice comme la suivante :

$$C(X, Y) = \begin{pmatrix} \frac{1}{x_1 - y_1} & \dots & \frac{1}{x_1 - y_n} \\ \vdots & \frac{1}{x_i - y_j} & \vdots \\ \frac{1}{x_n - y_1} & \dots & \frac{1}{x_n - y_n} \end{pmatrix}$$

**Remarque 6.** La condition que l'on a prise plus haut permet d'avoir des coefficients toujours bien définis.

La matrice de Cauchy apparaît lors de la recherche de fractions rationnelles sous contraintes. On fixe les pôles simples que la fraction doit admettre. Les coefficients solutions sont alors solution du système défini par la matrice de Cauchy.

- (b) Une matrice de Hankel (H) est une matrice carrée dont les coefficients sont constants le long des diagonales ascendantes.

Les coefficients  $H_{ij}$  de la matrice vérifient alors la relation :

$$H_{ij} = H_{i-1,j+1}$$

Soit  $H \in \mathbb{M}_n(\mathbb{K})$  alors  $H$  à  $2n - 1$  diagonales. On obtient donc une matrice comme la suivante :

Soit  $(x_0, \dots, x_{2n-2}) \in \mathbb{K}^{2n-1}$

$$H = \begin{pmatrix} x_0 & \dots & \dots & x_{n-3} & x_{n-2} & x_{n-1} \\ \vdots & & \ddots & x_{n-2} & x_{n-1} & x_n \\ \vdots & \ddots & \ddots & \ddots & x_n & x_{n+1} \\ x_{n-3} & x_{n-2} & \ddots & \ddots & \ddots & \vdots \\ x_{n-2} & x_{n-1} & x_n & \ddots & & \vdots \\ x_{n-1} & x_n & x_{n+1} & \dots & \dots & x_{2n-2} \end{pmatrix}$$

- (c) Une matrice de Toeplitz (T) est une matrice carrée dont les coefficients sont constants le long des diagonales.

Les coefficients  $T_{ij}$  de la matrice vérifient alors la relation :

$$T_{ij} = T_{i+1,j+1}$$

Soit  $T \in \mathbb{M}_n(\mathbb{K})$  alors  $T$  à  $2n - 1$  diagonales. On obtient donc une matrice comme la suivante :

Soit  $(x_0, \dots, x_{2n-2}) \in \mathbb{K}^{2n-1}$

$$T = \begin{pmatrix} x_0 & x_n & x_{n+1} & \dots & \dots & x_{2n-2} \\ x_1 & x_0 & x_n & \ddots & & \vdots \\ x_2 & x_1 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & x_n & x_{n+1} \\ \vdots & & \ddots & x_1 & x_0 & x_n \\ x_{n-1} & \dots & \dots & x_2 & x_1 & x_0 \end{pmatrix}$$

La matrice de Toeplitz apparaît souvent dans les problèmes de traitement du signal et de l'image, mais aussi en physique statistique. En effet, elle intervient dans la résolution du modèle d'Ising sous certaines hypothèses simplificatrices. Ce dernier permet d'approcher l'évolution de phénomènes macroscopiques causés par l'interaction de particules binaires. Il est notamment utilisé pour la modélisation de phénomènes ferromagnétiques. La matrice de Toeplitz est si courante car elle est caractéristique d'une invariance par changement d'origine spatiale ou temporelle.

- (d) Une matrice de Vandermonde (V) est une matrice que l'on peut définir par un vecteur :

$$\alpha = (\alpha_1 \quad \dots \quad \alpha_n)^T$$

Les coefficients  $V_{ij}$  de la matrice vérifient alors la relation :

$$V_{ij} = \alpha_i^{j-1}$$

On obtient donc une matrice comme la suivante :

$$V(\alpha) = \begin{pmatrix} 1 & \alpha_1 & \alpha_1^2 & \dots & \alpha_1^{n-1} \\ 1 & \alpha_2 & \alpha_2^2 & \dots & \alpha_2^{n-1} \\ 1 & \alpha_3 & \alpha_3^2 & \dots & \alpha_3^{n-1} \\ \vdots & \vdots & \vdots & \dots & \vdots \\ 1 & \alpha_n & \alpha_n^2 & \dots & \alpha_n^{n-1} \end{pmatrix}$$

La matrice de Vandermonde apparaît lorsque l'on cherche le polynôme de degré  $n - 1$  qui passe par  $n$  points donnés du plan, d'abscisses toutes distinctes. Elle intervient avec le polynôme d'interpolation de Lagrange.

En effet, en prenant les points  $(\alpha_i, y_i)$  tel que  $\forall i \neq j, \alpha_i \neq \alpha_j$ , si on écrit le polynôme d'interpolation comme suit :

$$P_n(x) = \sum_{i=1}^n \omega_i x^{i-1} \quad \text{tel que} \quad \forall i \quad P_n(\alpha_i) = y_i$$

alors :

$$V(\alpha) \begin{pmatrix} \omega_1 \\ \omega_2 \\ \vdots \\ \omega_n \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}$$

## 2.2 Déplacement et rang de déplacement

On voit donc que les différentes matrices structurées présentées se caractérisent par le fait que leurs coefficients peuvent être définis par un nombre de paramètre ( $\mathcal{O}(n)$ ) bien inférieur à  $n^2$  et ayant peu de coefficients non nuls, c'est à dire étant denses. On peut alors espérer exploiter cette particularité pour diminuer le nombre d'opération requises pour les traiter, en *compressant* ces matrices. C'est à dire parvenir à réduire l'espace occupé par ces matrices sans pour autant perdre l'information qu'elles portent. Il peut également s'avérer utile de se munir d'une sorte de *distance*, qui pourrait nous indiquer si une matrice est plus ou moins structurée et de quel type de structure elle dispose. C'est à l'ensemble de ces problèmes et interrogations que la notion de déplacement apporte une réponse.

Il convient de rappeler que dans le cas particulier des matrices de Toeplitz, des algorithmes rapides de factorisations étaient déjà connus. Le premier a été introduit par Levinson en 1947 [Lev47], puis des améliorations ont régulièrement été apportées jusqu'aux années 60. Cependant, aucun de ces algorithmes n'utilisaient la notion de déplacement. Nous allons voir que nous ne réalisons complètement notre objectif de compression qu'en utilisant la notion de déplacement. Cette dernière a été introduite aux alentours de 1980 dans les articles de Friedlander et al puis dans Kailath et al.

L'heuristique des méthodes de factorisation qui utilisent le déplacement est la suivante :

Comme on vient de le rappeler les matrices *structurées* sont denses, mais dépendent de peu de paramètres. On va donc leur appliquer une transformation pour les rendre creuses et considérablement diminuer leur rang. On peut alors à partir de là obtenir une représentation efficace et compacte sur laquelle l'implémentation des algorithmes est moins coûteuse. Enfin, on peut remonter à la solution voulue après le travail sur la forme compacte. On peut de manière schématique voir ces étapes comme une *compression* et un *traitement* de données compressées puis un retour à la représentation initiale que l'on peut voir comme une *décompression*. Bien sûr, pour que cette méthode ait un quelconque intérêt, il est nécessaire que nous soyons en mesure d'effectuer cette phase de *décompression* et que le coût des étapes de *compression* puis *décompression* ne soit pas supérieur au coup des méthodes classiques.

La notion de déplacement a d'abord été très liée aux matrices de Toeplitz. En effet, elle s'est développée comme une formalisation des idées apparues dans [Lev47]. Elle a ensuite été étendue aux autres types de matrices que nous avons introduites plus haut. La notion de déplacement est portée par les opérateurs de déplacements que nous allons maintenant introduire.

**Définition 2.3.** Soit  $A \in \mathbb{M}_n(\mathbb{K})$

Alors le spectre de  $A$  est l'ensemble des valeurs propres de  $A$ . On le note  $Sp(A)$ .

**Définition 2.4.** Soient  $M \in \mathbb{M}_n(\mathbb{K})$  et  $N \in \mathbb{M}_n(\mathbb{K})$  tels que  $Sp(M) \cap Sp(N) = \emptyset$

On définit l'opérateur de déplacement  $\nabla$  par :

$\forall A \in \mathbb{M}_n(\mathbb{K})$ ,

$$\nabla_{M,N}(A) = MA - AN \quad (9)$$

**Remarque 7.** Il est courant de trouver dans la littérature une définition qui ne comporte pas de conditions sur les spectres. Cependant, comme nous le verrons plus loin, cette condition est nécessaire et suffisante pour assurer l'inversibilité de l'opérateur. Sachant que nous ne nous concentrerons que sur des cas où l'opérateur est inversible - ce qui est absolument nécessaire pour l'étape de "décompression" - nous avons fait le choix d'inclure cette condition dans la définition.

**Définition 2.5.** Soient  $M \in \mathbb{M}_n(\mathbb{K})$  et  $N \in \mathbb{M}_n(\mathbb{K})$

On définit l'opérateur de déplacement  $\Delta$  par :

$\forall A \in \mathbb{M}_n(\mathbb{K})$ ,

$$\Delta_{M,N}(A) = A - MAN \quad (10)$$

L'opérateur  $\nabla$  est dérivé d'une équation de Sylvester, c'est à dire une équation matricielle du type :

$$AX - XB = Y$$

Mentionnons que  $A$  et  $B$  et  $X$  sont des matrices de  $\mathbb{M}_n(\mathbb{C})$ . L'équation de Sylvester est un outil dans de nombreux domaines. En effet, savoir quand cette équation admet une unique solution permet de dégager des résultats sur la similarité et la commutativité des matrices, sur les sous-espaces hyper-invariants ou encore sur certaines équations différentielles.

L'opérateur  $\Delta$  est quant à lui dérivé d'une équation de Stein, c'est à dire du type :

$$Y = X - AXB$$

Dans notre cas, nous travaillerons quasi-exclusivement avec l'opérateur  $\nabla$ . En effet, sa condition d'inversibilité est simple à manipuler. De plus, l'opérateur  $\Delta$  n'a pas de conditions d'inversibilité générale. Il nécessite une étude au cas par cas pour chaque types de matrices structurées.

**Proposition 3.**  $\nabla_{M,N}$  est inversible si et seulement si  $Sp(M) \cap Sp(N) = \emptyset$

*Démonstration.* □

Comme nous l'avons mentionné plus haut, la notion de déplacement est d'abord apparue avec les matrices de Toeplitz mais s'est ensuite étendue aux autres types de matrices structurées. De ce fait, il existe un opérateur déplacement  $\nabla$  pour chaque type de matrices. L'équation reste la même, cependant les matrices  $M$  et  $N$  changent selon la structure que l'on étudie. Nous introduirons ici le terme de *matrices auxiliaires de déplacement*. Le couple  $(M, N)$  est donc caractéristique de chaque structures. Les transformations effectuées au moyen des matrices  $M$  et  $N$  doivent être *simple*. Elles sont donc essentiellement constituée de deux types de matrices. Nous allons à présents introduire ces deux types de matrices puis les opérateurs déplacements pour chaque type de structure.

**Définition 2.6.** Soit  $A \in \mathbb{M}_n(\mathbb{K})$

On nomme *matrices auxiliaires de déplacement*, les matrices que l'opérateur  $\nabla_{\bullet, \bullet}(A)$  prend en paramètre.

- (a) Le premier type de matrice auxiliaire de déplacement - que l'on notera  $Z_\beta$  - est une matrice de  $\mathbb{M}_n(\mathbb{K})$ . Elle effectue une permutation cyclique et une multiplication par un scalaire. Si on applique  $Z_\beta$  à droite (resp. à gauche) d'une matrice, toutes les lignes (resp. colonnes) de cette dernière sont décalées d'un cran vers le bas (resp. vers la droite), la dernière ligne (resp. première colonne) est remontée en première (resp. dernière) position et tout ses coefficients sont multipliés par un facteur  $\beta$ . On a alors :

$$Z_\beta = \begin{pmatrix} 0 & & & \beta \\ 1 & \ddots & & \\ & \ddots & \ddots & \\ & & & 1 & 0 \end{pmatrix} \quad (11)$$

Afin d'avoir une meilleure compréhension de l'action de  $Z_\beta$  on peut se donner un exemple en dimension réduite ( $n = 4$ ). Ici on ne traitera que le cas d'application à droite. En se munissant de scalaires quelconques, on obtient :

$$\begin{pmatrix} 0 & 0 & 0 & \beta \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \\ m & n & o & p \end{pmatrix} = \begin{pmatrix} \beta m & \beta n & \beta o & \beta p \\ a & b & c & d \\ e & f & g & h \\ i & j & k & l \end{pmatrix}$$

Dans tout les cas on peut résumer l'action de  $Z_\beta$  par les actions qui suivent. Quand on multiplie par  $Z_\beta$  à droite :

$$\begin{aligned} L_i &\leftarrow L_{i-1} \\ L_1 &\leftarrow \beta L_n \end{aligned}$$

où  $L_i$  est la  $i^{\text{ème}}$  ligne de la matrice multipliée.

Quand  $Z_\beta$  est appliquée à gauche :

$$\begin{aligned} C_i &\leftarrow C_{i+1} \\ C_n &\leftarrow \beta C_1 \end{aligned}$$

où  $C_i$  est la  $i^{\text{ème}}$  colonne de la matrice multipliée.

- (b) Le second type de matrice auxiliaire de déplacement est une matrice diagonale de  $\mathbb{M}_n(\mathbb{K})$ . Elle prend en entrée les coordonnées d'un vecteur de  $\mathbb{K}^n$  et les répartit sur sa diagonale. On la nomme  $D$ . Si on applique  $D$  à droite (resp. à gauche) d'une matrice alors chaque ligne (resp. colonne) de la matrice est multipliée par la coordonnée de  $X$  qui lui correspond. Il s'agit en quelques sorte d'une mise à l'échelle.

Soit  $X \in \mathbb{K}^n$  tel que  $X = (x_1, \dots, x_n)$  alors on a :

$$D(X) = \text{diag}(x_1, \dots, x_n) = \begin{pmatrix} x_1 & & & \\ & \ddots & & \\ & & \ddots & \\ & & & x_n \end{pmatrix} \quad (12)$$

Afin d'avoir une meilleure compréhension de l'action de  $D$  on peut se donner un exemple en dimension réduite ( $n = 4$ ). Ici on ne traitera que le cas d'application à droite. En se munissant de scalaires quelconques, on obtient :

$$\begin{pmatrix} x_1 & 0 & 0 & 0 \\ 0 & x_2 & 0 & 0 \\ 0 & 0 & x_3 & 0 \\ 0 & 0 & 0 & x_4 \end{pmatrix} \begin{pmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \\ m & n & o & p \end{pmatrix} = \begin{pmatrix} x_1a & x_1b & x_1c & x_1d \\ x_2e & x_2f & x_2g & x_2h \\ x_3i & x_3j & x_3k & x_3l \\ x_4m & x_4n & x_4o & x_4p \end{pmatrix}$$

Dans tout les cas on peut résumer l'action de  $D$  par les actions qui suivent.  
Quand  $D$  est appliquée à droite :

$$L_i \leftarrow x_i L_i$$

où  $L_i$  est la  $i^{\text{ème}}$  ligne de la matrice multipliée.

Quand  $D$  est appliquée à gauche :

$$C_i \leftarrow x_i C_i$$

où  $C_i$  est la  $i^{\text{ème}}$  colonne de la matrice multipliée.

Ces matrices auxiliaires de déplacement sont donc relativement *simple*, dans la mesure où elles ne font intervenir que des permutations de lignes entre elles et de colonnes entre elles ainsi que des multiplications par un scalaire. Outre cela et c'est probablement leur caractéristique la plus importante, elle sont creuse. On peut donc aisément calculer leur inverse pour un coût moindre. Cela est un réel atout pour la phase de *décompression*.

Nous allons maintenant nous atteler à fournir des exemples de l'effet de l'opérateur de déplacement sur les matrices structurées. Nous traiterons ici les cas des matrices de Cauchy, Hankel, Toeplitz et Vandermonde.

(a) En conservant les notations précédentes, pour une matrice de Cauchy on a :

$$\begin{aligned} \nabla_{D(X),D(Y)}(C(X,Y)) &= D(X)C(X,Y) - C(X,Y)D(Y) \\ &= \begin{pmatrix} x_1 & & & \\ & x_2 & & \\ & & \ddots & \\ & & & x_n \end{pmatrix} \begin{pmatrix} \frac{1}{x_1-y_1} & \cdots & \frac{1}{x_1-y_n} \\ \vdots & \frac{1}{x_i-y_j} & \vdots \\ \frac{1}{x_n-y_1} & \cdots & \frac{1}{x_n-y_n} \end{pmatrix} - \begin{pmatrix} \frac{1}{x_1-y_1} & \cdots & \frac{1}{x_1-y_n} \\ \vdots & \frac{1}{x_i-y_j} & \vdots \\ \frac{1}{x_n-y_1} & \cdots & \frac{1}{x_n-y_n} \end{pmatrix} \begin{pmatrix} y_1 & & & \\ & y_2 & & \\ & & \ddots & \\ & & & y_n \end{pmatrix} \\ &= \begin{pmatrix} 1 & 1 & \cdots & 1 \\ 1 & 1 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \cdots & 1 \end{pmatrix} \end{aligned}$$

(b) De même pour une matrice de Hankel on a :

$$\nabla_{Z_0,Z_0^T} = Z_0 H - H Z_0^T$$

$$\begin{aligned}
&= \begin{pmatrix} 0 & & & 0 \\ 1 & \ddots & & \\ & \ddots & \ddots & \\ & & 1 & 0 \end{pmatrix} \begin{pmatrix} x_0 & \dots & x_{n-2} & x_{n-1} \\ \vdots & \ddots & \ddots & x_n \\ x_{n-2} & \ddots & \ddots & \vdots \\ x_{n-1} & x_n & \dots & x_{2n-2} \end{pmatrix} - \begin{pmatrix} x_0 & \dots & x_{n-2} & x_{n-1} \\ \vdots & \ddots & \ddots & x_n \\ x_{n-2} & \ddots & \ddots & \vdots \\ x_{n-1} & x_n & \dots & x_{2n-2} \end{pmatrix} \begin{pmatrix} 0 & 1 & & \\ & \ddots & \ddots & \\ & & \ddots & 1 \\ 0 & & & 0 \end{pmatrix} \\
&= \begin{pmatrix} 0 & \dots & 0 & 0 \\ x_0 & \dots & x_{n-2} & x_{n-1} \\ \vdots & \ddots & \ddots & \vdots \\ x_{n-2} & x_{n-1} & \dots & x_{2n-3} \end{pmatrix} - \begin{pmatrix} 0 & x_0 & \dots & x_{n-2} \\ \vdots & \vdots & \ddots & x_{n-1} \\ 0 & x_{n-2} & \ddots & \vdots \\ 0 & x_{n-1} & \dots & x_{2n-3} \end{pmatrix} \\
&= \begin{pmatrix} 0 & -x_0 & \dots & -x_{n-2} \\ x_0 & & & \\ \vdots & & & \\ x_{n-2} & & & \end{pmatrix}
\end{aligned}$$

(c) De même pour une matrice de Toeplitz on a :

$$\begin{aligned}
&\nabla_{Z_1, Z_0}(T) = Z_1 T - T Z_0 \\
&= \begin{pmatrix} 0 & & & 1 \\ 1 & \ddots & & \\ & \ddots & \ddots & \\ & & 1 & 0 \end{pmatrix} \begin{pmatrix} x_0 & x_n & \dots & x_{2n-2} \\ x_1 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & x_n \\ x_{n-1} & \dots & x_1 & x_0 \end{pmatrix} - \begin{pmatrix} x_0 & x_n & \dots & x_{2n-2} \\ x_1 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & x_n \\ x_{n-1} & \dots & x_1 & x_0 \end{pmatrix} \begin{pmatrix} 0 & & & 0 \\ 1 & \ddots & & \\ & \ddots & \ddots & \\ & & \ddots & 1 \\ & & & 0 \end{pmatrix} \\
&= \begin{pmatrix} x_{n-1} & x_{n-2} & \dots & x_0 \\ & & & x_{2n-2} \\ & * & & \vdots \\ & & & x_n \end{pmatrix} - \begin{pmatrix} x_n & x_{n+1} & \dots & 0 \\ & & & \vdots \\ & * & & \vdots \\ & & & 0 \end{pmatrix} \\
&= \begin{pmatrix} x_{n-1} - x_n & x_{n-2} - x_{n+1} & \dots & x_0 \\ & & & x_{2n-2} \\ & & & \vdots \\ & & & x_n \end{pmatrix}
\end{aligned}$$

Le symbole \* dénote la même sous-matrice dans les deux matrices où il apparaît.

(d) De même pour une matrice de Vandermonde on a :

$$\begin{aligned}
&\nabla_{D(\alpha), Z_0}(V(\alpha)) = D(\alpha)V(\alpha) - V(\alpha)Z_0 \\
&= \begin{pmatrix} \alpha_1 & & & \\ & \alpha_2 & & \\ & & \ddots & \\ & & & \alpha_n \end{pmatrix} \begin{pmatrix} 1 & \alpha_1 & \dots & \alpha_1^{n-1} \\ \vdots & \alpha_2 & \dots & \alpha_2^{n-1} \\ \vdots & \vdots & & \vdots \\ 1 & \alpha_n & \dots & \alpha_n^{n-1} \end{pmatrix} - \begin{pmatrix} 1 & \alpha_1 & \dots & \alpha_1^{n-1} \\ \vdots & \alpha_2 & \dots & \alpha_2^{n-1} \\ \vdots & \vdots & & \vdots \\ 1 & \alpha_n & \dots & \alpha_n^{n-1} \end{pmatrix} \begin{pmatrix} 0 & & & 0 \\ 1 & \ddots & & \\ & \ddots & \ddots & \\ & & \ddots & 1 \\ & & & 0 \end{pmatrix}
\end{aligned}$$

$$\begin{aligned}
&= \begin{pmatrix} \alpha_1 & \alpha_1^2 & \dots & \alpha_1^n \\ & * & & \alpha_2^n \\ & & & \vdots \\ & & & \alpha_n^n \end{pmatrix} - \begin{pmatrix} \alpha_1 & \dots & \alpha_1^{n-1} & 0 \\ & & & \vdots \\ & * & & \vdots \\ & & & 0 \end{pmatrix} \\
&= \begin{pmatrix} & & & \alpha_1^n \\ & & & \alpha_2^n \\ & & & \vdots \\ & & & \alpha_n^n \end{pmatrix}
\end{aligned}$$

Le symbole  $*$  dénote la même sous-matrice dans les deux matrices où il apparaît.

On peut alors constater que le rang de la matrice obtenue après application de l'opérateur de déplacement est très faible. Dans le cas de la matrice de Cauchy, le rang est égal à 1. Il est inférieur ou égal à 2 dans le cas de la matrice de Toeplitz ainsi que dans le cas de la matrice de Hankel et inférieur ou égal à 1 dans le cas de la matrice de Vandermonde (en fonction de la valeur des coefficients). Cette propriété de réduction drastique du rang nous sera très utile lors de la phase de *compression*.

**Définition 2.7.** Soit  $A \in \mathbb{M}_n(\mathbb{K})$

Alors on appelle rang de  $A$ , la dimension de l'image de  $A$ . On note :

$$rg(A) = \dim \text{Im}(A)$$

**Définition 2.8.** Soit  $A \in \mathbb{M}_n(\mathbb{K})$

Alors on appelle rang de déplacement de  $A$  la rang de l'image de  $A$  par l'opérateur de déplacement  $\nabla$ . On note :

$$r = rg(\nabla_{M,N}(A))$$

On peut résumer les différents rangs de déplacement dans le tableau suivant :

Structure	Matrice	$M$	$N$	$rg(\nabla_{M,N}(A))$
Cauchy	$C(X, Y)$	$D(X)$	$D(Y)$	$= 1$
Hankel	$H$	$Z_0$	$Z_0^T$	$\leq 2$
Toeplitz	$T$	$Z_1$	$Z_0$	$\leq 2$
Vandermonde	$V$	$D(\alpha)$	$Z_0$	$\leq 1$

### 2.3 Propriétés de l'opérateur de déplacement

Outre son action sur le rang, le déplacement s'avère digne d'intérêt en pratique du fait de ses nombreuses propriétés de *conservation*. C'est grâce à ces dernières que nous pourrions mettre en œuvre les méthodes de factorisation sans que les propriétés de structure ne soient modifiées. Ce sont également ces propriétés qui nous permettront d'effectuer la phase de *décompression*, c'est à dire remonter des générateurs vers la matrice d'intérêt.

**Proposition 4.** Soit  $A, B, L \in \mathbb{M}_n(\mathbb{K})$ , alors on a :

- (i)  $\nabla_{M,N}(A + B) = \nabla_{M,N}(A) + \nabla_{M,N}(B)$
- (ii)  $\nabla_{M,N}(A^T) = -\nabla_{N^T,M^T}(A)^T$
- (iii)  $\nabla_{M,N}(A^{-1}) = -A^{-1}\nabla_{N,M}(A)A^{-1}$
- (iv)  $\nabla_{M,N}(AB) = \nabla_{M,L}(A)B + A\nabla_{L,N}(B)$

*Démonstration.*

(i)

$$\begin{aligned}\nabla_{M,N}(A + B) &= M(A + B) - (A + B)N \\ &= MA - AN + MB - BN \\ &= \nabla_{M,N}(A) + \nabla_{M,N}(B)\end{aligned}$$

(ii)

$$\begin{aligned}\nabla_{M,N}(A^T) &= MA^T - A^T N \\ &= -(A^T N - MA^T) \\ &= -[(N^T A)^T - (AM^T)^T] \\ &= -(N^T A - AM^T)^T \\ &= -\nabla_{N^T,M^T}(A)^T\end{aligned}$$

(iii)

$$\begin{aligned}\nabla_{M,N}(A^{-1}) &= MA^{-1} - A^{-1}N \\ &= -A^{-1}N + MA^{-1} \\ &= -A^{-1}(N - AMA^{-1}) \\ &= -A^{-1}(NA - AM)A^{-1} \\ &= -A^{-1}\nabla_{N,M}(A)A^{-1}\end{aligned}$$

(iv)

$$\begin{aligned}\nabla_{M,N}(AB) &= MAB - ABN \\ &= MAB - ALB + ALB - ABN \\ &= (MA - AL)B + A(LB - BN) \\ &= \nabla_{M,L}(A)B + A\nabla_{L,N}(B)\end{aligned}$$

□

### 3 Exploitation de la structure de déplacement : méthodes de factorisation rapide

#### 3.1 Factorisation de rang plein et générateurs

Dans toute la suite de ce chapitre on se fondera essentiellement sur l'article [?]. On expose la factorisation de rang plein, ou factorisation de rang maximale comme suit.

**Théorème 5.** *Soit  $A \in \mathbb{M}_n(\mathbb{K})$  tel que  $rg(A) > 0$   
Alors on peut écrire  $A$  comme un produit de matrices,*

$$A = GH^T$$

Où  $G \in \mathbb{M}_{n \times l}(\mathbb{K})$  et  $H \in \mathbb{M}_{n \times l}(\mathbb{K})$  tel que l'on ait

$$rg(G) = rg(H) = l$$

*Démonstration.* Soit  $G \in \mathbb{M}_{n \times l}(\mathbb{K})$  tel que ses colonnes forment une base de l'espace des colonnes de  $A$ . Alors tout les vecteurs de l'espace des colonnes de  $A$  peuvent être exprimé comme une combinaison linéaire des colonnes de  $G$ . En particulier chaque colonnes  $\mathbf{a}_i$  de  $A$  peut être écrite comme :

$$\exists \mathbf{v}_i \in \mathbb{R}^l, \forall i \in \llbracket 1, n \rrbracket$$

$$\mathbf{a}_i = G\mathbf{h}_i,$$

Donc il existe une matrice  $H \in \mathbb{M}_{n \times l}$  telle que

$$A = GH^T = G(\mathbf{h}_1; \dots; \mathbf{h}_n)$$

De plus, on a :

$$l = rg(A) = rg(GH^T) \leq rg(H^T) \leq l$$

on conclut donc que  $H$  est bien de rang maximal pour ces colonnes.  $\square$

**Définition 3.1.** *En restant dans le cadre du Théorème 5, on nomme le couple  $\{G, H^T\}$  les générateurs de  $A$ .*

On peut ici prendre conscience du phénomène de compression. En effet, on passe d'une matrice  $A$  ayant  $n^2$  coefficients à un produit de matrices ayant  $2 \times l \times n$  coefficients. Lorsque  $n$  est très grand, comme c'est très souvent le cas, cette différence est fortement significative. Le nombre d'entrée à donc été fortement réduit et ceux sans perdre aucune information car on a bien égalité entre les deux membres. Dans la pratique, on applique la factorisation aux matrices déplacées, c'est à dire aux matrices structurées aux quelles on a appliqué l'opérateur  $\nabla$ .

Si nous nous trouvons en présence d'une matrice déplacée de type Toeplitz, alors en procédant par identification on obtient ces générateurs de la manière suivante :

$$\forall i \in \llbracket 2, n \rrbracket \quad g_{i,1} = a_{i-1,n} + a_{i,1} \quad g_{i,2} = 0 \quad g_{1,1} = g_{1,2} = 1 \forall j \in \llbracket 2n - 1 \rrbracket \quad (13)$$

Les générateur sont également digne d'intérêt car grâce aux conditions que nous nous sommes donné sur l'opérateur de déplacement dans le second chapitre, dans certains cas, nous sommes en mesure de remonter à la matrice originale à partir de la donnée des générateur  $\{G, H^T\}$  et des matrices auxiliaires de déplacement  $M$  et  $N$  de l'opérateur  $\nabla$ .

**Proposition 5.** *Soit  $A \in \mathbb{M}_n(K)$  Si  $A$  est une matrice de type Cauchy de générateur  $\{G, H^T\}$  alors on a :*

$$A = \sum_{i=1}^r D(\mathbf{G}_i)C(X, Y)D(\mathbf{H}_i)$$

où  $\mathbf{G}_i$  et  $\mathbf{H}_i$  sont respectivement les  $i^{\text{ème}}$  colonnes de  $G$  et de  $H$  et  $X$  et  $Y$  des vecteurs de  $\mathbb{K}^n$ .

### 3.2 Algorithme GKO

## A Boîte à outils

Dans cette partie nous introduirons les propriétés dont nous aurons besoin pour développer notre travail.

**Définition A.1.** *On nomme l'ensemble des matrices triangulaires supérieures de  $\mathbb{K}$  de dimension  $n$  :  $\mathbb{T}_n(\mathbb{K})$*

*On rappelle que le produit et la somme de matrices triangulaires restent une matrice triangulaire.*

**Proposition 6.** *L'inverse d'une matrice triangulaire supérieure (resp. inférieure) est une matrice triangulaire supérieure (resp. inférieure).*

*Démonstration.* On se concentrera ici sur le cas des matrices triangulaires supérieures, la démonstration est en tout points similaire pour les matrices triangulaires inférieures.

Soit  $A \in \mathbb{T}_n(\mathbb{K})$ , telle que  $A$  soit inversible

On considère

$$\begin{aligned} f_A : \mathbb{T}_n(\mathbb{K}) &\longrightarrow \mathbb{T}_n(\mathbb{K}) \\ M &\longmapsto AM \end{aligned}$$

Soient  $\lambda \in \mathbb{K}$  et  $N \in \mathbb{T}_n(\mathbb{K})$  alors,

$$f_A(M + \lambda N) = A(M + \lambda N) = AM + \lambda AN = f_A(M) + \lambda f_A(N)$$

$f_A$  est donc linéaire, de plus on a

$$f_A(M) = 0 \Leftrightarrow AM = 0 \Leftrightarrow A^{-1}AM = 0 \Leftrightarrow M = 0$$

Donc  $f_A$  est injective. On se trouve en dimension finie avec égalité des espaces de départ et d'arrivée, on peut donc appliquer le théorème du rang. On en déduit donc que  $f_A$  est inversible. Or  $I_n \in \mathbb{T}_n(\mathbb{K})$  donc  $\exists X \in \mathbb{T}_n(\mathbb{K})$  tel que  $f_A(X) = I_n$  c'est à dire tel que  $AX = XA = I_n$ , il s'agit donc de l'inverse de  $A$  qui appartient à  $\mathbb{T}_n(\mathbb{K})$ , on sait que cet inverse est unique.  $\square$

## Références

- [GPS90] K. A. Gallivan, R. J. Plemmons, and A. H Sameh. Parallel algorithms for dense linear algebra computations. *SIAM Rev.*, 32(1) :54–135, 1990.
- [KKM79] T. Kailath, S.-Y. Kung, and M. Morf. Displacement ranks of matrices and linear equations. *J. Math. Anal. Appl.*, 68(2) :395–407, 1979.
- [Lev47] N. Levinson. The wiener rms error criterion in filter design and prediction. *J. Math. Phys.*, 25(4) :261–278, 1947.