# Méthodes itératives pour la résolution d'équations matricielles

El Mostafa Sadek

UNIVERSITÉ CADI AYYAD
FACULTÉ DES SCIENCES
ET TECHNIQUES
MARRAKECH

UNIVERSITÉ DU LITTORAL.
CÔTE D'OPALE, LMPA.
- CALAIS -
FRANCE.

# THÈSE DE DOCTORAT

Préparée dans le cadre d'une cotutelle entre
l'Université de Cadi Ayyad et l'Université du Littoral

**Spécialité : Mathématiques Appliquées et Informatique**

**CED : Sciences de l'Ingénieur**

---

# Méthodes itératives pour la résolution d'équations matricielles.

---

Thèse présentée par : **SADEK EL MOSTAFA**
Soutenue publiquement, le 23 Mai 2015 devant le jury :

**Rapporteurs :**

| | |
|---|---|
| Abouir Jilali | Professeur, FST Université Hassan II, Mohammedia, Maroc |
| El Hajji Said | Professeur, Université Mohamed V, Rabat, Maroc |
| Guedda Mohammed | Professeur, Université de Picardie Jules Verne, Amiens France |

**Examinateur :**

| | |
|---|---|
| Sadok Hassane | Professeur, Université du Littoral Côte d'Opale, France |

**Directeurs de Thèse :**

| | |
|---|---|
| Bentbib Abdeslem Hafid | Professeur, FST, Université Cadi Ayaad, Marrakech |
| Jbilou Khalide | Professeur, Université du Littoral Côte d'Opale, France |

Cette thèse a été préparée aux :

Laboratoire de Mathématiques Appliquées et Informatique
Université Cadi Ayyad
Faculté des Sciences et Techniques.
B.P. 549, Av. Abdelkarim Elkhattabi
Guéliz Marrakech - Maroc
☎ 05 24 43 34 04 / 05 24 43 31 63
**Fax :** 05 24 43 31 70
**Site :** www.fstg-marrakech.ac.ma

Laboratoire de mathématiques pures et appliquées
Centre Universitaire de la Mi-Voix
Maison de la Recherche Blaise Pascal
50, Rue Ferdinand Buisson
CS 80699
Calais Cedex, France
☎ 03 21 46 55 90
**fax :** 03 21 46 55 86
**Site :** www-lmpa.univ-littoral.fr

# FICHE PRESENTATIVE DE LA THESE

- **Nom et Prénom de l'auteur** : SADEK EL MOSTAFA
- **Intitulé du travail** : Méthodes itératives pour la résolution d'équations matricielles.

**Encadrants** :
- Abdeslem Hafid BENTBIB, Professeur de l'enseignement supérieur
- **Laboratoire** : LAMAI, Laboratoire de Mathématiques Appliquées et Informatique.

- Khalide JBILOU, Professeur de l'enseignement supérieur
- **Laboratoire** : LMPA, Laboratoire de Mathématiques Pures et Appliquées.

- **Lieux de réalisation des travaux** :
• Faculté des Sciences et Techniques, Laboratoire de Mathématiques Appliquées et Informatique, Marrakech.
• Université du Littoral Côte d'Opale, Laboratoire de Mathématiques Pures et Appliquées, Calais France.

- **Période de réalisation du travail de thèse** : 4 ans

- **Rapporteurs de thèse** :

| | |
|---|---|
| Abouir Jilali | Professeur, FST Université Hassan II, Mohammedia, Maroc |
| El Hajji Said | Professeur, Université Mohamed V, Rabat, Maroc |
| Guedda Mohammed | Professeur, Université de Picardie Jules Verne, Amiens, France. |

- **Cadres de coopération** : Cotutelle de thèse avec l'université de Littorale Côte d'Opale, France
- **Ce travail a donné lieu aux résultats suivants** :

## Liste des publications

☛ S. Agoujil, Abdeslem Hafid Bentbib, Khalide Jbilou and EL Mostafa Sadek. *A Minimal residal norm method for large-scale Sylvester matrix equations*. Electronic Transactions on Numerical Analysis. Volume 43, pp. 45–59, 2014.
☛ Abdeslem Hafid Bentbib, Khalide Jbilou and EL Mostafa Sadek. *On some Krylov subspace based methods for large-scale nonsymmetric algebraic Riccati problems*, soumis.
☛ Abdeslem Hafid Bentbib, Khalide Jbilou and EL Mostafa Sadek. *A minimal residual method for large scale Riccati matrix equations*, en préparation.

## Liste des communications

☛ *Méthode itérative pour la résolution de l'équation de Riccati*, au congrès international de la Société Marocaine de Mathématiques Appliquées (SM2A), organisé à Marrakech en 2012.

☛ *A minimal residual method for large scale Sylvester matrix equations*, au congrès international : Numerical Analysis and Scientific Computition with Applications (NASCA13), organisé à Calais France en 2013.

☛ *A new projection method for solving large-scale nonsymmetric algebric Riccati equations* , au congrès international JANO à EST ESSAOUIRA en 2013.

☛ *A Minimal residal norm method for large-scale Sylvester matrix equations*, au congrès international : Modélisation et Calcul Scientifique pour L'Ingénierie Mathématiques MOCASIMà Marrakech en 2014.

# Table des matières

**Table des Matières** **4**

**Liste des Tableaux** **7**

**Liste des Figures** **8**

**Notations** **9**

**Introduction Générale** **10**

**1 Outils de développement théoriques** **13**
1.1 Introduction . . . 13
1.2 Matrices particulières . . . 13
1.3 Norme de Frobenius . . . 14
1.4 Formule de Sherman-Morrison Woodbury . . . 14
1.5 Inverse généralisée d'une matrice . . . 14
1.6 La décomposition en valeurs singulières (SVD) . . . 15
1.7 Approximation de rang inférieur . . . 16
1.8 Le produit de Kronecker et le $\diamond$-produit . . . 16
1.9 Méthodes de Krylov . . . 18
1.9.1 Sous-espace de Krylov par blocs . . . 18
1.9.2 Sous-espace de Krylov étendu par blocs . . . 19
1.9.3 Sous-espace de Krylov étendu global . . . 22
1.9.4 Algorithme d'Arnoldi étendu global . . . 23
1.10 Contrôlabilité et l'observabilité . . . 25

**2 Méthodes de type globale-minimisation pour les équations de Lyapunov de grandes dimensions** **27**
2.1 Introduction . . . 27
2.2 Solution exacte de l'équation de Lyapunov . . . 29
2.3 Méthode de Galerkin . . . 30
2.4 Méthode de minimisation du résidu . . . 33
2.5 Méthodes itératives pour résoudre le problème réduit . . . 34
2.5.1 LSQR global . . . 34
2.5.2 La méthode du gradient conjugué globale préconditionné . . . 38
2.6 Exemples numériques . . . 41
2.6.1 Exemple 1 . . . 42
2.6.2 Exemple 2 . . . 43

2.7 Conclusion . . . . . 44

**3 L'Extended-Bloc Arnoldi algorithme avec minimisation de résidu pour les équations de Sylvester creuse et de grandes taille 45**
3.1 Introduction . . . . . 45
3.2 Méthode de type Galerkin . . . . . 47
3.3 Méthode de minimisation du résidu . . . . . 48
3.4 Méthodes pour résoudre le problème de minimisation réduit . . . . . 50
3.4.1 La méthode directe basée sur la formulation de Kronecker . . . . . 50
3.4.2 Méthode de Hu et Reichel . . . . . 51
3.4.3 La Méthode LSQR globale . . . . . 54
3.4.4 La méthode du gradient conjugué globale préconditionné . . . . . 56
3.5 Forme factorisée de l'approximation de la solution . . . . . 58
3.6 L'algorithme GA-LRSE . . . . . 59
3.7 L'algorithme MR-LR-Sylvester . . . . . 60
3.8 Résolution de l'équation de Stein non symétrique . . . . . 60
3.8.1 Méthode de Galerkin . . . . . 62
3.8.2 Méthode de minimisation du résidu . . . . . 65
3.9 Exemples numériques . . . . . 66
3.9.1 Tests numérique pour l'équation de Sylvester . . . . . 67
3.9.2 Tests numérique pour l'équation de Stein non symétrique . . . . . 72
3.10 Conclusion . . . . . 74

**4 Méthode de minimisation du résidu pour la résolution de l'équation de Riccati continue de grande taille 76**
4.1 Introduction . . . . . 76
4.2 Méthode de Galerkin . . . . . 78
4.3 Méthode de minimisation du résidu . . . . . 79
4.4 Résolution du problème de minimisation réduit . . . . . 81
4.5 Méthode de MINRES globale. . . . . 84
4.6 Forme factorisée de la solution approchée . . . . . 88
4.7 Algorithme GA-CAREs . . . . . 89
4.8 Algorithme MR-CAREs . . . . . 89
4.9 Exemples numériques . . . . . 90
4.9.1 Exemple 1 . . . . . 91
4.9.2 Exemple 2 . . . . . 93
4.9.3 Exemple 3 . . . . . 93
4.10 Conclusion . . . . . 93

**5 Équation matricielle de Riccati non symétrique et application à l'équation de transport 95**
5.1 Introduction . . . . . 95
5.2 La méthode de Newton-Krylov par blocs . . . . . 98
5.3 Solution approchée de rang inférieur de NAREs . . . . . 100
5.4 Applications à la théorie de transport . . . . . 106
5.5 Résultats numériques . . . . . 108
5.6 Conclusion . . . . . 110

**Bibliographie** **112**

**Résumé** **120**

**Abstract** **122**

# Liste des tableaux

3.1 Résultats pour l'exemple 1.6 . . . . . 71
3.2 Résultats de l'exemple 1.7. . . . . 72
3.3 Résultats de l'exemple 2.4 . . . . . 74

4.1 Résultats de l'exemple 2 . . . . . 93
4.2 Résultats de l'exemple 3. . . . . 93

5.1 Résultats pour les exemples de NAREs dans la théorie de transport avec $c = 0.5$ et $\alpha = 0.5$. . . . . 109
5.2 Résultats pour les exemples de NAREs dans la théorie de transport avec $c = 0.9999$ et $\alpha = 10^{-8}$. . . . . 109
5.3 Résultats pour les exemples 4, comparaisons avec la méthode SDA. . . . . 110

# Table des figures

2.1 Résultats de l'exemple 1 : MR(PCGG) . . . . . 42
2.2 Résultats de l'exemple 1 : MR(GL-LSQR) . . . . . 43
2.3 Résultats de l'exemple 2 : MR(PCGG) . . . . . 43
2.4 Résultats de l'exemple 2 : MR(GL-LSQR) . . . . . 44

3.1 Résultat pour l'exemple 1.1. . . . . 68
3.2 Résultat pour l'exemple 1.2. . . . . 68
3.3 Résultat pour l'exemple 1.3. . . . . 69
3.4 Résultat pour l'exemple 1.4. . . . . 69
3.5 Résultat pour l'exemple 1.5. . . . . 70
3.6 Résultats de l'exemple 2.1 . . . . . 72
3.7 Résultats de l'exemple 2.2 . . . . . 73
3.8 Résultats de l'exemple 2.3 . . . . . 73

4.1 Valeurs singulières de la solution exacte. . . . . 88
4.2 Résultats de l'exemple 1.1 . . . . . 92
4.3 Résultats de l'exemple 1.2. . . . . 92

5.1 Les valeurs singulières de la solution non négative minimale de l'équation NAREs . . . . . 101

# Table des notations

| | |
|---|---|
| $\mathbb{R}$ | le corps des nombres réels. |
| $\mathbb{C}$ | le corps des nombres complexes. |
| $\mathbb{R}^{m\times n}$ | l'espace des matrices à $m$ lignes et $n$ colonnes, à coefficients dans $\mathbb{R}$. |
| $I_n$ | la matrice identité de taille $n \times n$. |
| $I$ | $I_n$ s'il n'y a pas de confusion. |
| $0_n$ | la matrice nulle d'ordre $n$. |
| $e_k$ | la $k^{ième}$ colonne de $I$. |
| $A^{-1}$ | l'inverse de la matrice $A$. |
| $sp(A)$ | le spectre de la matrice $A$. |
| $A^T$ | la matrice transposée de $A$. |
| $A^{-T}$ | $(A^{-1})^T = (A^T)^{-1}$. |
| $det(A)$ | le déterminant de la matrice $A$. |
| $\lambda(A)$ | valeur propre de la matrice $A$. |
| $x^T$ | le transposé du vecteur $x$. |
| $\otimes$ | le produit de Kronecker. |
| $\diamond$ | le produit de diamond. |
| $\mathcal{K}_m(A, V)$ | le sous espace de Krylov par blocs. |
| $\mathcal{K}_m^e(A, V)$ | le sous espace de Krylov étendu par blocs. |
| $\mathbb{K}_m^e(A, V)$ | le sous espace de Krylov étendu global. |
| $Sep_F(A_1, A_2)$ | la séparation entre $A_1$ et $A_2$, $Sep_F(A_1, A_2) = \min_{\|X\|_F=1} \|A_1X - XA_2\|$. |
| $<,>$ | le produit scalaire usuel. |
| $<,>_F$ | le produit scalaire de Frobenius. |
| $X \perp_F Y$ | $< X, Y >_F = 0$. |
| $\|x\|$ | la norme euclidienne du vecteur $x$. |
| $\|A\|_F$ | la norme de Frobenius de la matrice $A$. |
| $\delta_{i,j}$ | le symbole de Kronecker. |
| $\square$ | fin de la démonstration. |

# Introduction Générale

Les équations matricielles interviennent dans de nombreux domaines des mathématiques, des sciences physiques, dans différents domaines des sciences de l'ingénieur, traitement du signal, restauration d'images, filtrage, réduction de modèle en théorie du contrôle, contrôle optimal, discrétisation d'équations aux dérivées partielles, et la résolution d'équations différentielles ordinaires ou aux dérivées partielles, la théorie de transport, voir par exemple les références suivantes [21, 39, 56, 62].

En analyse numérique, pour résoudre les équations algébriques matricielles, il existe plusieurs méthodes possibles : certaines sont directes et d'autres itératives. Les méthodes directes sont très efficaces : elles donnent la solution exacte (aux erreurs d'arrondi près) du système linéaire considéré (ou d'une équation matricielle). Pour des problèmes de très grande taille, cependant ces méthodes directes peuvent devenir très chères, elles ont l'inconvénient de nécessiter une assez grande place mémoire car elles nécessitent le stockage de toute la matrice en mémoire vive. Les méthodes itératives de type Krylov, employées depuis une trentaine d'années, sont généralement plus efficaces dans les problèmes de grande taille. La plupart des méthodes itératives manipulent le système linéaire au travers de produits "matrice-vecteur", ce qui réduit la place mémoire.

Les équations algébriques matricielles ont été obtenues à partir de la discrétisation d'équations aux dérivés partielles, elles sont en général de grande taille. Les méthodes itératives basées sur une technique de projection sur un sous-espaces de Krylov de dimension $m$ (ou sur des sous-espaces de Krylov étendus) sont plus efficaces et rapides (la vitesse de convergence). Ces méthodes peuvent être classées en deux catégories principales : les méthodes de projections basées sur la condition de Petrov-Galerkin [62, 98, 100, 102] et les méthodes du résidu minimal MR (Minimal Residual) [74, 82, 94, 95, 99, 100] (par exemple GMRES, MINRES).

La résolution d'équations matricielles telles que l'équation de Sylvester [13, 18, 21, 39, 56, 57, 74], Lyapunov [15, 60, 61, 68, 98, 102] ou celle de Riccati [20, 55, 64, 65] a connu un développement considérable ces vingt dernières années. Ceci est du à leurs applications dans de nombreux domaines : le contrôle optimal, le traitement de signal et la

restauration d'images, réduction de modèle en théorie du contrôle, calcul d'un contrôle optimal en contrôle linéaire quadratique. Le cas d'équations de Riccati non symétriques jouent aussi un rôle important dans de nombreuses applications dont principalement : la théorie de transport [25, 38, 71, 72], théorie des jeux [1].

Pour des matrices creuses et de grandes tailles, peu de méthodes efficaces existent. Les méthodes itératives de type de projection sur des sous-espaces de Krylov sont généralement plus efficaces et rapides. Nous proposons dans ce travail de thèse d'étudier de nouvelles méthodes itératives de type de projection sur des sous espaces de Krylov étendu (extended Krylov subspace) [36] pour résoudre les équations matricielles suivantes : Lyapunov, Sylvester, Stein, Riccati symétrique et Riccati non symétrique.

Cette thèse se décompose de la façon suivante :

Le premier Chapitre sera consacré aux rappels de plusieurs définitions, théorèmes et des rappels des sous-espaces de type Krylov qui seront utilisés tout au long de cette thèse.

Le deuxième Chapitre portera sur les équations algébriques de Lyapunov $AX + XA^T + BB^T = 0$, où $A$ matrice de grande taille et $B$ une matrice de rang inférieur. Nous proposons deux méthodes itératives de projection sur des sous espaces de Krylov étendu global. La première basée sur la condition d'orthogonalité de Galerkin et la deuxième avec la condition de minimisation de résidu. Nous donnerons des résultats de majoration de l'erreur et l'efficacité des méthodes proposées par des tests numériques.

Dans le Chapitre 3, nous nous intéresserons à la résolution numérique des équations de Sylvester $AX + XB = EF^T$, où les matrices $A$ et $B$ sont de grandes tailles et creuses et les matrices $E$ et $F$ sont de rang inférieur. Nous proposons une nouvelle méthode de projection sur des sous espaces de Krylov étendu par blocs. Cette méthode est basée sur la condition de minimisation du résidu MR (Minimal Residual) [74, 82, 94, 99, 100]. Le problème réduit obtenu est résolu soit par des méthodes directes (méthode de Hu et Reichel [57]) ou bien des méthodes itératives : Gl-LSQR, PGCG. Les approximations de la solution sont données sous forme factorisée, nous permettant d'économiser de la place mémoire. Des exemples numériques sont présentés montrant l'efficacité de la méthode proposée.

Le Chapitre 4 porte sur la résolution d'équations matricielles non linéaires de Riccati symétriques $A^TX + XA - XBB^T + CC^T = 0$, qui intervient dans plusieurs problèmes, notamment dans les problèmes de calcul de contrôle linéaire quadratique. Dans ce chapitre, nous appliquons la méthode de minimisation du résidu MR (Minimal Residual) à l'équation de Riccati. Cette méthode est basée sur des sous espaces de Krylov étendu par blocs, l'algorithme d'Arnoldi étendu et la méthode MINRES [94] pour résoudre le problème de minimisation projetée. Les solutions approchées (symétrique) sont données

sous la forme $X_m = Z_m Z_m^T$, où $Z_m$ une matrice de rang inférieur. Enfin, ce chapitre se termine par des exemples numériques pour valider notre approche.

Le dernier chapitre de cette thèse, concerne le cas de l'équation de Riccati non symétrique NARE (Nonsymmetric Algebraic Riccati Equation) $XCX - XD - AX + B = 0$. Nous proposons deux nouvelles méthodes itératives pour résoudre NARE. La première est une méthode de type Newton-Krylov, basée sur la méthode de Newton et des sous-espaces de Krylov par blocs : à chaque itération de la méthode de Newton, on résout une équation de Sylvester de grande taille par l'utilisation de l'algorithme d'Arnoldi par blocs. La deuxième méthode est une méthode itérative de projection sur des sous espaces de Krylov étendus par blocs avec la condition d'orthogonalité de Galerkin. Nous allons également présenter des résultats théoriques de la majoration de l'erreur. La performance des approches proposées au regard d'autres méthodes est confirmée par des tests comparatifs.

Chapitre 1

# Outils de développement théoriques

## 1.1 Introduction

Dans ce chapitre, nous introduisons les notations et les définitions qui seront utilisées tout au long de cette thèse. Nous donnons quelques définitions importantes par la suite : matrices particulières, la décomposition en valeurs singulières d'une matrice , puis l'inverse généralisée d'une matrice. Nous rappelons ensuite le produit de Kronecker et le $\diamond$-produit [12, 30]. Enfin, nous donnons un rappel sur des sous espaces de type Krylov [36, 64, 65, 100, 101] et des algorithmes pour construire une base orthonormée des sous espaces de Krylov, voir [55, 56, 101, 102] pour plus de détails.

## 1.2 Matrices particulières

Soit $A = (a_{ij})$ une matrice carrée de taille $n \times n$.

**Définition 1.2.1.** *La matrice $A$ est dite :*
- *symétrique si $A^T = A$,*
- *diagonale si $a_{ij} = 0$ pour $i \neq j$,*
- *triangulaire inférieure si $a_{ij} = 0$ pour $i < j$,*
- *triangulaire supérieure si $a_{ij} = 0$ pour $i > j$,*
- *orthogonale si $A^T A = AA^T = I_n$,*
- *Hessenberg supérieure si $a_{ij} = 0$ pour $i > j + 1$,*
- *définie positive si $(Ax, x) > 0$ pour tout $x \in \mathbb{R}^n / \{0\}$,*
- *non négative si tous les éléments $a_{ij}$ de $A$ sont positifs et $a_{ij} \neq 0$,*
- *Une matrice $A$ dont tous les termes extra-diagonaux sont négatifs ou nuls est appelée $Z$-matrice.*
- *$A$ est une $M$-matrice si $A$ est une $Z$-matrice inversible et non négative*

## 1.3 Norme de Frobenius

Pour deux matrices $Y$ et $Z \in \mathbb{R}^{n\times s}$, le produit scalaire de Frobenius de $Y$ et $Z$ est défini par

$$\langle Y, Z\rangle_F = tr(Y^T Z),$$

où $tr(Y^T Z)$ désigne la trace de la matrice $Y^T Z$. La norme associée est celle de Frobenius noté par $\| \, . \, \|_F$, et définie par

$$\| A \|_F = tr(A^T A) = \sum_{i}^{n}\sum_{j}^{m}(a_{i,j})^2, \text{ où } \; A = [a_{i,j}] \in \mathbb{R}^{n\times m}.$$

## 1.4 Formule de Sherman-Morrison Woodbury

**Proposition 1.4.1.** *Soient $B$ une matrice inversible d'ordre $n$ et $u, v \in \mathbb{R}^n$. Alors la matrice $B + uv^T$ est inversible si et seulement si $1 + v^T B^{-1} u \neq 0$. Dans ce cas, nous avons la formule suivante :*

$$(B + uv^T)^{-1} = (I - \frac{(B^{-1}u)v^T}{1 + v^T B^{-1} u})B^{-1} \tag{1.1}$$

La formule généralisée, connue sous le nom de formule de Sherman-Morrison-Woodbury, est définie dans la proposition suivante.

**Proposition 1.4.2.** *Soient $A \in \mathbb{R}^{n\times n}$ et $C \in \mathbb{R}^{m\times m}$ deux matrices inversibles. Soient $U \in \mathbb{R}^{n\times m}$ et $V \in \mathbb{R}^{m\times n}$. Si la matrice $(C^{-1}+VA^{-1}U)$ est inversible, alors $(A+UC^{-1}V)$ est inversible, et vérifie la formule de Sherman-Morrison-Woodbury donnée comme suit :*

$$(A + UC^{-1}V)^{-1} = A^{-1} - A^{-1}U(C^{-1} + VA^{-1}U)^{-1}VA^{-1}$$

## 1.5 Inverse généralisée d'une matrice

**Définition 1.5.1.** *La pseudo-inverse, $A^+$, d'une matrice $A$ de taille $m\times n$, à coefficients réels ou complexes est l'unique matrice de taille $n \times m$ satisfaisant les quatre critères suivants :*

1. $AA^+A = A$,
2. $A^+AA^+ = A^+$,
3. $(AA^+)^T = AA^+$ *($AA^+$ est une matrice symétrique),*
4. $(A^+A)^T = A^+A$ *($A^+A$ est également symétrique).*

**Remarque 1.5.2.** *Pour les matrices dont les composantes sont des nombres complèxes au lieu des nombres réels, $A^T$ sera remplacée par $A^*$, où $A^*$ désigne l'adjointe de la matrice A. La pseudo-inverse appelée aussi inverse au sens de Moore-Penrose existe et unique pour toute matrice réelles ou complexe.*

Nous donnons la proposition suivante

**Proposition 1.5.3** ([43]). *Le pseudo-inverse d'une matrice $A \in \mathbb{R}^{m\times n}$ de rang maximal est définie par :*
– $A^+ = A^T(AA^T)^{-1}$ *si* $rang(A) = m$,
– $A^+ = (A^TA)^{-1}A^T$ *Si* $rang(A) = n$.

## 1.6 La décomposition en valeurs singulières (SVD)

En mathématiques, le procédé d'algèbre linéaire de décomposition en valeurs singulières (ou la décomposition SVD : Singular Value Decomposition) d'une matrice est un outil important de factorisation de matrices rectangulaires réelles ou complexes, elle est l'une des méthodes de factorisation la plus générale et la plus utile. La décomposition en valeurs singulières est utilisée, entre autres, dans le calcul de pseudo-inverse, dans la résolution de problèmes aux moindres carrés et dans les méthodes de régularisation. On l'emploie également en traitement d'image, en traitement du signal, et pour les approximations de rang inférieur d'une matrice.

**Théorème 1.6.1** (Golub et Van Loan, 1990). *Soit A une matrice réelle de dimension $m \times n$. Il existe deux matrices orthogonales U et V respectivement de dimension $m \times m$ et $n \times n$ tel que*

$$A = U\Sigma V^T, \tag{1.2}$$

*où $\Sigma = diag[\sigma_1, \sigma_2, ..., \sigma_m] \in \mathbb{R}^{m\times n}$. Les $\sigma_i$ sont appellés les valeurs singulières de A et elles vérifient $\sigma_1 \geq \sigma_2 \geq ... \geq \sigma_r > \sigma_{r+1} = ... = \sigma_m = 0$, $r \leq \min\{m, n\}$. Les vecteurs $u_i$ (colonnes de U) et $v_i$ (colonnes de V) sont appellés les vecteurs singuliers de gauche respectivement de droite.*

**Remarque 1.6.2.** *Le rang de la matrice A est égal au nombre de valeurs singulières non nulles que possède la matrice A.*
*La décomposition SVD d'une matrice A de taille $m \times n$ nécessite $4m^2n + 8mn^2 + 9n^3$ opérations élémentaires. Avec MATLAB on obtient la décomposition singulière avec la commande[U,S,V]=svd(A).*

Nous donnons le théorème suivant

**Théorème 1.6.3** ([43]). *Soit $A \in \mathbb{R}^{m\times n}$ une matrice de rang r, et soit sa décomposition en valeurs singulières $A = U\Sigma V^T$, où $\Sigma = diag[\sigma_1, \sigma_2, ..., \sigma_r, 0, ..., 0] \in \mathbb{R}^{m\times n}$. Alors le*

*pseudo-inverse de $A$ est donné par :*

$$A^+ = V\Sigma^+ U^T,$$

*où* $\Sigma^+ = diag[1/\sigma_1, 1/\sigma_2, ..., 1/\sigma_r, 0, ..., 0] \in \mathbb{R}^{n\times m}$.

## 1.7 Approximation de rang inférieur

Soit $A \in R^{m\times n}$ une matrice. Déterminer une matrice $X$ de même taille que $A$ mais de rang inférieur $k$ de telle sorte que $\|A - X\|_F$ soit minimale, est un problème classique. La solution de ce problème est donnée par le théorème d'Eckart-Young suivent

**Théorème 1.7.1** (Eckart et Young)**.** *Soit $A$ une matrice de rang $r$, et soit sa décomposition en valeurs singulières*

$$A = U\Sigma V^T,$$

*où* $\Sigma = diag[\sigma_1, \sigma_2, ..., \sigma_r, 0, ..., 0] \in \mathbb{R}^{m\times n}$*, et soit $k$ un entier inférieur ou égal à $r$. Si on note* $\Sigma_k = diag[\sigma_1, \sigma_2, ..., \sigma_k]$*, alors nous avons*

$$\min_{rang(X)\leq k} \|A - X\|_F = \|A - A_k\|_F = \left( \sum_{i=k+1}^{r} \sigma_i^2(A) \right)^{\frac{1}{2}},$$

*où*

$$A_k = U \begin{pmatrix} \Sigma_k & 0 \\ 0 & 0 \end{pmatrix} V^T.$$

Ce théorème établit une relation entre le rang $k$ de l'approximation $X$ de $A$ et les valeurs singulières $\sigma_{k+1}, ..., \sigma_r$ de $A$. Par conséquent, si les valeurs singulières décroissent rapidement vers zéro alors nous pouvons espérer déterminer une approximation de $A$ possédant un rang très faible.

## 1.8 Le produit de Kronecker et le $\diamond$-produit

Le produit de Kronecker est un outil important dans l'algèbre linéaire et les équations matricielles. Il permet de transformer une équation matricielle en un système linéaire, pour plus de détails voir par exemple le livre [96]. Le $\diamond$-produit est défini dans [12, 30], c'est un outil important pour résoudre les équations matricielles par les méthodes de type Krylov global, voir [12, 30].

**Définition 1.8.1.** *Soient $A$ et $B$ deux matrices de taille $m \times n$ et $s \times t$ respectivement. Le produit de Kronecker $A \otimes B$ est la matrice de taille $ms \times tn$ définie par :*

$$A \otimes B = \begin{pmatrix} a_{1,1}B & \cdots & \cdots & a_{1,n}B \\ a_{2,1}B & \ddots & & a_{2,n}B \\ \vdots & & \ddots & \vdots \\ a_{m,1}B & a_{m,2}B & \cdots & a_{m,n}B \end{pmatrix}$$

Pour $Y = [y_{i,j}] \in \mathbb{R}^{n\times s}$; on note par $vec(Y)$ le vecteur de $\mathbb{R}^{ns}$ défini par $vec(Y) = [y(.,1)^T, (y.,2)^T, ..., y(.,s)^T]$ où $y(.,j), j = 1, ..., s$ est la $j$ème colonne de $Y$.

On a les propriétés suivantes :

**Proposition 1.8.2** ([12, 30])**.** *Soient $A$ et $B$ deux matrices de taille $m \times n$ et $s \times t$ réspctivement. Nous avons les proprétés suivantes :*

1. $(A \otimes B)^T = A^T \otimes B^T$.
2. $(A \otimes B)(C \otimes D) = (AC \otimes BD)$.
3. *Si $A$ et $B$ sont des matrices non singulières de dimension $n \times n$ et $p \times p$, respectivement, alors $(A \otimes B)^{-1} = A^{-1} \otimes B^{-1}$.*
4. *Si $A \in \mathbb{R}^{n\times n}$ et $B \in \mathbb{R}^{p\times p}$, alors*
   $det(A \otimes B) = det(A)^p det(B)^n$ *et* $tr(A \otimes B) = tr(A)tr(B)$.
5. $vec(ABC) = (C^T \otimes A)vec(B)$.
6. $vec(A)^T vec(B) = trace(A^T B)$.

On rappelle le produit diamond noté par $\diamond$ et qui est défini de la façon suivante :

**Définition 1.8.3** ([12, 30])**.** *Soient $A = [A_1, ..., A_p]$ et $B = [B_1, ..., B_l]$ deux matrices de taille $n \times ps$ et $n \times ls$, respectivement, où $A_i$ et $B_j (i = 1, ..., p; j = 1, ..., l)$ sont des matrices de $\mathbb{R}^{n\times s}$. Alors la matrice $A^T \diamond B \in \mathbb{R}^{p\times l}$ est définie par :*

$$A^T \diamond B = \begin{pmatrix} \langle A_1, B_1\rangle_F & \langle A_1, B_2\rangle_F & \cdots & \langle A_1, B_l\rangle_F \\ \langle A_2, B_1\rangle_F & & & \vdots \\ \vdots & & & \vdots \\ \langle A_p, B_1\rangle_F & \cdots & \cdots & \langle A_p, B_l\rangle_F \end{pmatrix}$$

**Remarque 1.8.4.**

1. *Si $s = 1$ alors $A^T \diamond B = A^T B$.*
2. *Si $s = 1, p = 1$ et $l = 1$, alors on pose $A = u \in \mathbb{R}^n$ et $B = v \in \mathbb{R}^n$, on aura $A^T \diamond B = u^T v \in \mathbb{R}$.*
3. *La matrice $A = [A_1, ..., A_p]$ est $F-$orthonormale si et seulement si $A^T \diamond A = I_p$.*
4. *Si $X \in \mathbb{R}^{n\times s}$, alors $X^T \diamond X = \| X \|_F^2$.*

On rappelle aussi les propriétés suivantes satisfaites par le $\diamond-$produit.

**Proposition 1.8.5** ([12, 30]). *Soient $A, B, C \in \mathbb{R}^{n\times ps}, D \in \mathbb{R}^{n\times n}, L \in \mathbb{R}^{p\times p}$ et $\alpha \in \mathbb{R}$. Alors on a :*

1. $(A+B)^T \diamond C = A^T \diamond C + B^T \diamond C$.
2. $A^T \diamond (B+C) = A^T \diamond B + A^T \diamond C$.
3. $(\alpha A)^T \diamond C = \alpha(A^T \diamond C)$.
4. $A^T \diamond B)^T = B^T \diamond A$.
5. $(DA)^T \diamond B = A^T \diamond (D^T B)$.
6. $A^T \diamond (B(L \otimes I_s)) = (A^T \diamond B)L$.
7. $\| A^T \diamond B \|_F \leq \| A \|_F \| B \|_F$.

## 1.9 Méthodes de Krylov

Les méthodes de projection sur les sous-espace de Krylov sont très utilisé dans dans des problèmes de grande dimension. Ces méthodes se basent sur des techniques de projection sur un sous-espace, appelé sous-espace de Krylov. Les méthodes itératives de type Krylov a connu un développement considérable dans les dernières années, pour résoudre les équations matricielles de grande taille.

Dans cette section, on va rappeler le sous-espace de Krylov par blocs, sous-espace de Krylov étendu par blocs ou global. On va rappeler aussi des algorithmes de type Arnoldi, pour construire une base orthogonale ou orthonormée d'un sous espace de type Krylov.

### 1.9.1 Sous-espace de Krylov par blocs

Le sous espace de Krylov par blocs associé à la paire $(A, V)$ est défini par

$$\mathbb{K}_m(A, V) = Image(\{V, AV, A^2V, \cdots, A^{m-1}V\}).$$

L'algorithme d'Arnoldi par blocs ci-dessus consiste à construire une base orthonormale de l'espace $\mathbb{K}_m(A, V)$.

**Algorithm 1** Algorithme d'Arnoldi par blocs

– Calculer la décomposition $QR$ de $V$ : $[V_1, R]$ =qr$(V)$
– **Pour** $j = 1 : m$
– $W = A\,V_j$
– **Pour** $i = 1, \ldots, j$
– $H_{i,j} = V_i^T\,W$
– $W = W - V_i\,H_{i,j}$
– **Fin Pour** $i$
– Calculer la décomposition QR de $W$ : $[V_{j+1}, H_{j+1,j}]$ =qr$(W)$
– **Fin Pour** $j$

On note $\mathcal{V}_m$ la matrice réelle de taille $n \times mr$ définie par

$$\mathcal{V}_m = [V_1, \ldots, V_m]$$

et $\tilde{\mathcal{H}}_m$ la matrice réelle de taille $(m+1)r \times mr$ de type Hessenberg supérieure par blocs, dont les blocs non nuls $H_{i,j}$ de taille $r \times r$ sont produits par l'algorithme précédent. On vérifie les identités suivantes

$$\begin{aligned} A\,\mathcal{V}_m &= \mathcal{V}_m\,\mathcal{H}_m + V_{m+1}\,H_{m+1,m}\,E_m^T \\ &= \mathcal{V}_{m+1}\tilde{\mathcal{H}}_m, \end{aligned}$$

où $E_m = [0_r, \ldots, 0_r, I_r]$ est la matrice de taille $mr \times r$ composée des $r$ dernières colonnes de la matrice identité $I_{mr}$ et $\mathcal{H}_m$ est la matrice de taille $mr \times mr$ obtenue en supprimant les $r$ dernières lignes de $\tilde{\mathcal{H}}_m$.

### 1.9.2 Sous-espace de Krylov étendu par blocs

Pour la résolution des systèmes linéaires, il existe plusieurs méthodes possibles ; certaines sont directes et d'autres itératives. Dans des problèmes de très grande taille, ces méthodes peuvent devenir très chères, et sont limitées par la taille du problème et bien que les dernières versions de Matlab prennent en charge des problèmes de taille allant jusqu'à $\mathcal{O}(10^4)$, le temps CPU ( Central Processing Unit) ainsi que la taille mémoire peuvent poser problème. Les méthodes efficaces sont les méthodes de sous-espace Krylov et consistent à construire une suite de sous espaces emboîtés de taille croissante qui vont contenir à la limite la solution recherchée. Elles sont basées sur une technique de projection sur un sous espace de Krylov, de dimension $m$ ($m \leq n$) plus petite que la taille $n$ du problème. Parmi les nombreux ouvrages de référence traitant des méthodes de Krylov, on pourra citer le livre de Yousef Saad [100]. Ces méthodes diffèrent l'une de l'autre par le type de projection qui est appliquée et le choix des sous-espaces de Krylov.

Dans cette section, on va rappeler le sous-espace de Krylov étendu par blocs.

Supposons que nous calculons des approximations des expressions de la forme $w := f(A)v$, où $f$ est une fonction non linéaire définie sur $A$.
Dans [36], Druskin et Knizhnerman ont montré que $f$ ne peut pas être approchée avec précision par un polynôme de degré $m-1$. Pour cette raison, ils ont proposé l'utilisation de la méthode de sous-espace de Krylov étendu, ce qui permet d'approcher $f$ par une fonction rationnelle.
Le sous-espace de projection que l'on considère peut être considéré comme une somme de deux sous-espaces de Krylov par blocs. Le premier lié à $\mathbb{K}_m(A,V)$ et le deuxième à $\mathbb{K}_m(A^{-1},V)$. Plus précisément, en supposant que la matrice $A$ est inversible, le sous-espace de Krylov étendu par blocs associé à $(A,V)$ est donné par [36, 102]

$$\begin{aligned}\mathcal{K}_m^e(A,V) &= Image\{V, A^{-1}V, AV, A^{-2}V, A^2V, \cdots, A^{m-1}V, A^{-m+1}V\}\\ &= \mathbb{K}_m(A,V) + \mathbb{K}_m(A^{-1}, A^{-1}V).\end{aligned}$$

nous avons,

$$\mathcal{K}_m^e(A,V) \subset \mathcal{K}_{m+1}^e(A,V) \;\text{ et }\; \mathcal{K}_m^e(A,V) \subset A\mathcal{K}_m^e(A,V).$$

Les méthodes de projection sont basées sur des algorithme qui permettent de construire une base orthonormée de l'espace de Krylov. parmi ces algorithmes, le plus connu est l'algorithme d'Arnoldi qui est basée sur d'orthonormalisation de Gram-Schmidt.

Pour construire une base orthonormée de l'espace de Krylov étendu $\mathcal{K}_m^e(A,V)$ en applique l'algorithme d'Arnoldi étendu par blocs ci-dessous ([55, 102] ).

Après $m$ étapes de l'algorithme d'Arnoldi étendu par blocs, on obtient la matrice

$$\mathbb{V}_m = [V_1, V_2, \ldots, V_m] \text{ où } V_i \in \mathbb{R}^{n\times 2r},$$

et les matrices de Hessenberg par blocs :

$$\overline{\mathbb{H}}_m = \begin{bmatrix} H_{1,1} & \cdots & \cdots & \cdots & H_{1,m} \\ H_{2,1} & \ddots & & & \vdots \\ 0 & \ddots & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & H_{m,m} \\ 0 & \cdots & \cdots & 0 & H_{m+1,m} \end{bmatrix}$$

**Algorithm 2** Algorithme d'Arnoldi étendu par blocs (EBA)

1. Entrées : les matrices $A \in \mathbb{R}^{n\times n}$ et $V \in \mathbb{R}^{n\times r}$ et $m$ entier.
2. Calculer la décomposition $QR$ de $[V, A^{-1}V]$, c'est-à-dire, $[V, A^{-1}V] = V_1\Lambda$ ;
3. On pose $\mathcal{V}_0 = [\,]$ ;
4. **Pour** $j = 1, 2, ..., m$ **faire**
   (a) On prend $V_j^{(1)} = V_j(:, 1:r)$ et $V_j^{(2)} = V_j(:, r+1:2r)$.
   (b) $\mathcal{V}_j = [\mathcal{V}_{j-1}, V_j]$ ; $\hat{V}_{j+1} = \left[A\,V_j^{(1)}, A^{-1}\,V_j^{(2)}\right]$ ;
   (c) Orthogonalisation de $\hat{V}_{j+1}$ avec les $\mathcal{V}_i$, $i = 1, 2, ..., j$, pour obtenir $V_{j+1}$, c'est-à-dire,
      i. **Pour** $i = 1, 2, \ldots, j$ **faire**
      ii. $H_{i,j} = V_i^T\,\hat{V}_{j+1}$ ;
      iii. $\hat{V}_{j+1} = \hat{V}_{j+1} - V_i\,H_{i,j}$ ;
      iv. **Fin Pour**
   (d) Calculer la décomposition $QR$ de $\hat{V}_{j+1}$, c'est-à-dire, $\hat{V}_{j+1} = V_{j+1}\,H_{j+1,j}$ ;
5. **Fin Pour**

et

$$\mathbb{H}_m = \begin{bmatrix} H_{1,1} & \cdots & \cdots & \cdots & H_{1,m} \\ H_{2,1} & \ddots & & & \vdots \\ 0 & \ddots & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & & 0 & H_{m,m-1} & H_{m,m} \end{bmatrix}$$

La matrice $\mathbb{H}_m$ de taille $2mr \times 2mr$ est obtenue en supprimant la dernière ligne par blocs de $\overline{\mathbb{H}}_m$. Par construction, la matrice $\mathbb{V}_m$ est orthonormale, ce qui signifie que les matrices $V_1, V_2, \cdots, V_m$ constituent une base orthonormale de l'espace de Krylov étendu $\mathcal{K}_m^e(A, V)$. Soit $\mathbb{T}_m \in \mathbb{R}^{2mr\times 2mr}$ la restriction de la matrice $A$ sur le sous-espace de Krylov étendu $\mathcal{K}_m^e(A, V)$, c'est-à-dire $\mathbb{T}_m = \mathbb{V}_m^T\,A\,\mathbb{V}_m$. Dans [102], Simoncini a démontré que la matrice $\mathbb{T}_m$ est également matrice de Hessenberg supérieure par blocs, et on peut calculer la matrice $\mathbb{T}_m$ à partir $\mathbb{H}_m$.
On établit les identités suivantes de manière immédiate

**Proposition 1.9.1.** *[55] Soit $\bar{\mathbb{T}}_m = \mathbb{V}_{m+1}^T A \mathbb{V}_m$, alors nous avons*

$$\begin{aligned} A\mathbb{V}_m &= \mathbb{V}_{m+1}\bar{\mathbb{T}}_m \\ &= \mathbb{V}_m\mathbb{T}_m + V_{m+1}T_{m+1,m}\mathbb{E}_m^T. \end{aligned}$$

*où $T_{i,j} \in \mathbb{R}^{2r\times 2r}$ est le bloc $(i, j)$ de la matrice $\mathbb{T}_m$ et $\mathbb{E}_m$ est la matrice de taille $mr \times r$ constituée des $r$ dernières colonnes de la matrice identité $I_{mr}$ (c.à.d $\mathbb{E}_m = [0_{2(m-1)r\times 2r}, I_{2r}]^T$).*

### 1.9.3 Sous-espace de Krylov étendu global

Le sous-espace de Krylov étendu global associé à $(A, V)$ est donne par [36, 56, 102]

$$\begin{aligned} \mathcal{K}_m^g(A,V) &= vect\{V, A^{-1}V, AV, A^{-2}V, A^2V, \cdots, A^{m-1}V, A^{-m+1}V\} \\ &= \mathbb{K}_m^g(A,V) + \mathbb{K}_m^g(A^{-1}, A^{-1}V). \end{aligned}$$

Pour construire une base une base orthonormale $\{V_1, V_2, ..., V_m\}$ de l'espace $\mathcal{K}_m^g(A,V)$, on a besoin de définir La factorisation $QR$ global.

#### La factorisation $QR$ global

Soit $Z = [Z_1, ..., Z_k] \in \mathbb{R}^{n\times ks}$ ( avec $Z_i \in \mathbb{R}^{n\times s}, i = 1, ..., k$). La décomposition globale de $QR$ de $Z$ est basée sur la procédure de Gram-Schmidt globale, elle permet de calculer une matrice $F$-orthonormale $Q = [Q_1, ..., Q_k]$ de taille $n \times ks$ telle que $vect\{Q_1, ..., Q_k\} = vect\{Z_1, ..., Z_k\}$ avec $\langle Q_i, Q_i\rangle_F = 1$ et $\langle Q_i, Q_j\rangle_F = 0$ si $i \neq j$.

L'algorithme est décrit ci-dessous :

**Algorithme 3 : L'algorithme de Gram-Schmidt modifié global (GGSG)**

**Entrée :** $Z = [Z_1, ..., Z_k]$ une matrice réelle de dimension $n \times sk$.
**Sortie :** $Q = [Q_1, ..., Q_k] \in \mathbb{R}^{n\times sk}$ F-orthonormale et $R \in \mathbb{R}^{k\times k}$ triangulaire supérieure.

1. $R = (r_{i,j}) = 0$.
2. $r_{1,1} =\| Z_1 \|_F$ .
3. $Q_1 = Z_1/r_{1,1}$.
4. **Pour** $i = 2, ..., k$, **faire :**
5. $\quad Q = Z_i$,
6. $\quad$ **Pour** $j = 1, ..., i-1$, **faire :**
7. $\quad\quad r_{j,i} = \langle Q, Z_j\rangle_F$,
8. $\quad\quad Q = Q - r_{j,i}Q_j$,
9. $\quad$ **Fin Pour** $j$.
10. $\quad r_{i,i} =\| Q \|_F$,
11. $\quad Q_i = Q/r_{i,i}$,
12. **Fin Pour** $i$.

**Proposition 1.9.2.** *[55] Soit $Z = [Z_1, ..., Z_k]$ une matrice de taille $n \times ks$ avec $Z_i \in \mathbb{R}^{n\times s}$, pour $i = 1, ..., k$. Donc en utilisant l'algorithme* 3*, la matrice $Z$ peut être factorisée sous la forme suivante*

$$Z = Q(R \otimes I_s),$$

*avec $Q = [Q_1, ..., Q_k]$ une matrice $F$-orthonormale de taille $n \times ks$ et qui satisfait $Q^T \diamond Q = I_k$ et $R$ une matrice triangulaire supérieure de taille $k \times k$.*

On note que $Q^T \diamond Z = Q^T \diamond (Q(R \otimes I_s))$, donc en utilisant la Proposition 1.8.5, on aura $Q^T \diamond Z = R$.

### 1.9.4 Algorithme d'Arnoldi étendu global

Dans ce paragraphe, on va décrire la procédure d'Arnoldi étendu global pour construire une base $F$-orthonormale $\{v_1, ..., v_{2m}\}$ de $\mathcal{K}_m^g(A, V)$ avec $v_i \in \mathbb{R}^{n\times s}$.
Les deux blocs $v_1$ et $v_2$ sont calculés de la manière suivante :

$$v_1 = \frac{V}{\omega_{1,1}}, \quad \texttt{et} \quad \omega_{2,2}v_2 = A^{-1}V - \omega_{1,2}v_1, \tag{1.3}$$

où les paramètres $\omega_{1,1}$ et $\omega_{2,2}$ sont tels que la matrice $[v_1, v_2]$ de dimension $n \times 2s$ est $F$-orthonormale. Donc $\omega_{1,1} =\| V \|_F$, $\omega_{1,2} = tr(V_1^T(A^{-1}V))$ et $\omega_{2,2} =\| U \|_F$, avec $U = A^{-1}V - \omega_{1,2}v_1$.
Pour calculer les blocs $v_{2j+1}$ et $v_{2j+2}$, pour $j = 1, ..., m-1$, on utilise les formules suivantes :

$$\begin{cases} h_{2j+1,2j-1}v_{2j+1} &= Av_{2j-1} - \sum_{i=1}^{2j} h_{i,2j-1}v_i, \\ h_{2j+2,2j}v_{2j+2} &= A^{-1}v_{2j} - \sum_{i=1}^{2j+1} h_{i,2j}v_i. \end{cases} \tag{1.4}$$

En utilisant les conditions d'orthogonalités $v_{2j+1} \perp_F v_1, ..., v_{2j}$ et $v_{2j+2} \perp_F v_1, ..., v_{2j+1}$, on obtient :

$$h_{i,2j-1} = tr(v_i^T A v_{2j-1}) = tr(v_i^T U_{i,1}^j) \quad \texttt{et} \quad h_{i,2j} = tr(v_i^T A^{-1} v_{2j}) = tr(v_i^T U_{i,2}^j),$$

où

$$U_{i,1}^j = Av_{2j-1} - \sum_{k=1}^{i-1} h_{k,2j-1}v_k \quad \texttt{et} \quad U_{i,2}^j A^{-1}v_{2j} - \sum_{k=1}^{i-1} h_{k,2j}v_k.$$

Les paramètres $h_{2j+1,2j-1}$ et $h_{2j+2,2j}$ sont tels que $\| v_{2j+1} \|_F = 1$ et $\| v_{2j+2} \|_F = 1$ respectivement. Donc,

$$h_{2j+1,2j-1} =\| U_{2j+1,1}^j \|_F \quad \texttt{et} \quad h_{2j+2,2j} =\| U_{2j+1,2}^j \|_F .$$

Maintenant, afin de décrire une nouvelle méthode pour calculer les vecteurs colonnes de la matrice $\mathbb{V}_m = [v_i]_{i=1,...,2m}$ de dimension $n \times 2ms$ et les entrées non nulles de la matrice

bloc de Hessenberg supérieure $H_m = [h_{i,j}]_{i=1,...,2m}^{j=1,...,2m}$, on introduit

$$V_i = [v_{2i-1}, v_{2i}] = [V_i^{(1)}, V_i^{(2)}], \quad \texttt{et} \quad H_{i,j} = \begin{pmatrix} h_{2i-1,2j-1} & h_{2i-1,2j} \\ h_{2i,2j-1} & h_{2i,2j} \end{pmatrix}, \tag{1.5}$$

qui sont, respectivement, le $i$ème vecteur bloc de dimension $n \times 2s$ de la matrice $\mathbb{V}_m$, et le $(i,j)$-ème bloc de taille $2 \times 2$ de la matrice bloc de Hessenberg supérieure $\mathbb{H}_m$. En utilisant le produit de Kronecker et les formules (1.4) et (1.5), on obtient

$$V_{j+1}(H_{j+1,j} \otimes I_s) = [AV_j^{(1)}, A^{-1}V_j^{(2)}] - \sum_{i=1}^{j} V_i(H_{i,j} \otimes I_s).$$

Comme les blocs $V_1, ..., V_m$ sont orthogonaux par rapport au $\diamond-$produit (i.e., $V_i^T \diamond V_j = 0_2$ pour $i \neq j$), alors en utilisant les propriétés du $\diamond-$produit, on trouve

$$H_{i,j} = V_i^T \diamond [AV_j^{(1)}, A^{-1}V_j^{(2)}] = V_i^T \diamond U_i^j, \quad i = 1, ..., j,$$

où $U_i^j = \sum_{k=1}^{i-1} V_k(H_{k,j} \otimes I_s)$. Pour les blocs triangulaires supérieurs $H_{j+1,j}$, on rappel que les deux matrices $H_{j+1,j}$ et $V_{j+1}$ sont calculées telles que $V_{j+1} \diamond V_{j+1} = I_2$. Donc on obtient $V_{j+1}$ et $H_{j+1,j}$ en calculant la factorisation $QR$ globale de $U_{j+1}^j$.
Soit $\Omega = [\omega_{i,j}]$ une matrice triangulaire supérieure de dimension $2 \times 2$ telle que ses entrées non nulles sont définies en (1.3), et on peut vérifier facilement que $\Omega$, et $V_1$ le premier bloc de la matrice $\mathbb{V}_m$ peuvent être obtenus facilement par la décomposition $QR$ globale de $[V, A^{-1}V]$.
Finalement, on peut résumer les résultats précédents dans l'algorithme suivant :

---

**Algorithme 4 : Algorithme d'Arnoldi étendu global (EGA)**

---

**Entrée :** $A \in \mathbb{R}^{n \times n}, V \in \mathbb{R}^{n \times s}$ et $m$ .
**Sortie :** $\mathbb{V}_m = [V_1, ..., V_m]$ de dimension $n \times 2ms$ et $\mathbb{H}_m \in \mathbb{R}^{2m \times 2m}$ matrice de Hessenberg supérieure.

1. On calcule la factorisation $QR$ globale de $[V, A^{-1}V]$, i.e , $[V, A^{-1}V] = V_1(\Omega \otimes I_s)$ ;
2. On pose $\mathbb{V}_0 = [\,]$ ;
3. **Pour** $j = 1, ..., m$, **faire :**
4. On prend $V_j^{(1)} = V_j(:, 1:s)$ ; $V_j^{(2)} = V_j(:, s+1:2s)$ ;
5. $\mathbb{V}_j = [\mathbb{V}_{j-1}, V_j]; U = [AV_j^{(1)}, A^{-1}V_j^{(2)}]$ ;
6. **Pour** $i = 1, ..., j$, **faire :**
7. $H_{i,j} = V_i^T \diamond U$ ;
8. $U = U - V_i(H_{i,j} \otimes I_s)$ ;
9. **Fin(i)**.

10. On calcule la décomposition $QR$ globale de $U$, i.e., $U = V_{j+1}(H_{j+1,j} \otimes I_s)$ ;
11. **Fin(j)**.

---

Si les matrices triangulaires supérieures $H_{j+1,j}$ ne sont pas de rang maximal, l'algorithme 1.3 calcule une matrice $F-$orthonormale $\mathbb{V}_m$ avec $\mathbb{V}_m = [V_1, ..., V_m]$ et $V_i \in \mathbb{R}^{n\times 2s}, (i = 1, ..., m)$ et $\overline{\mathbb{H}}_m$ une matrice bloc de Hessenberg supérieure de dimension $2(m+1) \times 2m$ avec $\overline{\mathbb{H}}_m = [H_{i,j}]$ et $H_{i,j} \in \mathbb{R}^{2\times 2}$.
On introduit une matrice de dimension $2m \times 2m$ donnée par $\mathbb{T}_m = \mathbb{V}_m^T \diamond (A\mathbb{V}_m) = [T_{i,j}]$, où $T_{i,j} = V_i^T \diamond (AV_j) \in \mathbb{R}^{2\times 2}$, pour $i, j = 1, ..., m$.
Soit $\overline{\mathbb{T}}_m = \mathbb{V}_{m+1}^T \diamond (A\mathbb{V}_m)$ et $E_m^T = [0_{2\times 2(m-1)}, I_2]$ une matrice qui correspond aux deux dernières colonnes de la matrice identité $2m \times 2m$, en utilisant le même argument utilisé dans le cas d'Arnoldi extended par bloc, on peut montrer qu'après $m$ itérations de l'algorithme 3, on aura :

$$A\mathbb{V}_m = \mathbb{V}_{m+1}(\overline{\mathbb{T}}_m \otimes I_s) = \mathbb{V}_m(\mathbb{T}_m \otimes I_s) + V_{m+1}(T_{m+1,m}E_m^T \otimes I_s).$$

La proposition suivante, donnée en [55], permet de calculer $\overline{\mathbb{T}}_m$ sans utiliser le produit matrice-vecteur avec $A$.

**Proposition 1.9.3.** *([55]) Soit $\overline{\mathbb{T}}_m = [t_{:,1}, ..., t_{:,2m}]$ et $\overline{\mathbb{H}}_m = [h_{:,1}, ..., h_{:,2m}]$ où $t_{:,i}, h_{:,i} \in \mathbb{R}^{2(m+1)}$ sont les $i$èmes colonnes de deux matrices blocs de Hessenberg supérieures $\overline{\mathbb{T}}_m$ et $\overline{\mathbb{H}}_m$ de dimension $2(m+1) \times 2m$. Donc les colonnes impaires sont telles que*

$$t_{:,2j-1} = h_{:,2j-1}, \quad \textit{pour} \quad j = 1, ..., m,$$

*tandis que les colonnes paires satisfont*

$$t_{:,2} = \frac{1}{\omega_{2,2}}(\omega_{1,1}e_1^{2(m+1)} - \omega_{1,2}h_{:,1}),$$

$$t_{:,2j+2} = \frac{1}{h_{2j+2,2j}}(e_{2j}^{2(m+1)} - t_{:,1:2j+1}h_{1:2j+1,2j}) \quad \textit{pour} \quad j = 1, ..., m-1,$$

*où $e_i^{(k)}$ est la $i$ème colonne de la matrice identité $I_k$ et $\omega_{1,1}, \omega_{1,2}$ sont déjà définis en* (1.3).

## 1.10 Contrôlabilité et l'observabilité

Dans cette partie, nous rappelons les résultats relatifs à la contrôlabilité et à l'observabilité de la paire $(A, B)$ et de la paire $(C, A)$ définies par un système dynamique linéaire à coefficients constants du type :

$$\begin{cases} \dot{x} = Ax + Bu, \\ y = Cx, \end{cases} \tag{1.6}$$

Ces notions sont largement développées dans [111]. Nous rappelons la notion de matrices stables dans la définition suivante :

**Définition 1.10.1.** *Une matrice $A$ est dite stable, si*

$$Sp(A) \subset \mathbb{C}^{-}.$$

**Définition 1.10.2.** *La paire $(A, B)$ est dite* **stabilisable** *s'il existe une matrice $F$ telle que $A + BF$ est stable.*

**Remarque 1.10.3.** *Un système dynamique linéaire à coefficients constants, donné par l'équation (1.6) est dit asymptotiquement stable, si $A$ est stable.*

Les définitions de la stabilisation et de la détectabilité du système linéaire (1.6) sont données dans [31].

**Définition 1.10.4.** *La paire $(C, A)$ est dite* **détectable** *si $(A^T, C^T)$ est stabilisable.*

Chapitre 2

# Méthodes de type globale-minimisation pour les équations de Lyapunov de grandes dimensions

## 2.1 Introduction

les équations matricielles de Lyapunov interviennent dans plusieurs domaines comme dans la théorie du contrôle [42], la théorie de la communication, l'analyse de la stabilité des systèmes dynamiques [87], la stabilité des équations différentielles ordinaires, la réduction de modèles [4, 5, 53], et la restauration d'images [18]. Par exemple, nous considérons le système linéaire dynamique $\sum$ invariant dans le temps :

$$\Sigma := \begin{cases} \dot{x}(t) = Ax(t) + Bu(t) \\ y(t) = Cx(t) \end{cases} \tag{2.1}$$

où $A$ est une matrice creuse de grande taille $n \times n$.

Le problème de réduction consiste à approcher $\Sigma$ par

$$\hat{\Sigma} := \begin{cases} \frac{d\hat{x}(t)}{dt} = \hat{A}\hat{x}(t) + \hat{B}u(t) \\ \hat{y}(t) = \hat{C}\hat{x}(t), \end{cases}$$

où $\hat{A} \in \mathbb{R}^{k\times k}$, $\hat{B} \in \mathbb{R}^{k\times s}$ et $\hat{C} \in \mathbb{R}^{k}$ avec $k << n$.

Parmi les méthodes de réduction des modèles de grande dimension, la plus connue est la troncation équilibrée. Cette méthode est basée sur le calcul des grammiens de commandabilité et d'observabilité. Les grammiens de commandabilité $W_c$ et d'observabilité

$W_o$ sont obtenus par résolution des équations de Lyapunov suivantes :

$$AW_c + W_cA^T + BB^T = 0,$$
$$A^TW_o + W_oA + C^TC = 0.$$

Il a été démontré que [42] :

$$\| \Sigma - \hat{\Sigma} \|_\infty \leq 2(\sigma_{k+1} + ... + \sigma_n)$$

où les $\sigma_i$ sont Les valeurs singulières de Hankel associées à $\Sigma$ donnée par : $\sigma_i = \sqrt{\lambda_i(W_cW_o)}$ arrangées en ordre décroissant, et la norme $\| \Sigma \|_\infty$ est la plus grande valeur singulière de la fonction de transfert associée à $\Sigma$ et définie par $\| \Sigma \|_\infty = \sup_{\omega\in\mathbb{R}}(\sigma_{max}(G(j\omega))$, où $G(.)$ représente la fonction de transfert associée à $\Sigma$ et $j\omega$ varie sur la totalité de l'axe imaginaire. Pour plus de détails voir [42, 68]. L'équation matricielle de Lyapunov est de la forme suivante

$$AX + XA^T + BB^T = 0, \tag{2.2}$$

où $A$, $B$ et $X$ sont des matrices respectivement de taille $n \times n$, $n \times s$, $n \times n$ avec $s << n$.

L'équation matricielle de Lyapunov peut être formulée comme un système linéaire de taille $n^2 \times n^2$. En utilisant le produit de Kronecker, nous obtenons

$$(I_n \otimes A + A \otimes I_n)vec(X) + vec(BB^T) = 0. \tag{2.3}$$

Donc l'équation de Lyapunov possède une solution unique $X$ si et seulement si $\lambda_i(A) + \lambda_j(A) \neq 0$ pour tout $i, j = 1, 2, \cdots, n$

Pour les équations matricielles de lyapunov de petite taille, il existe plusieurs méthodes dans la littérature, comme par exemple, l'algorithme de Bartels-Stewart [11], Hessenberg-Schur [45].

Pour la résolution de l'équation de Lyapunov (2.2), nous distinguons deux catégories. La première se sont Les méthodes directes, comme par exemple, l'algorithme de Bartels-Stewart [11], Hessenberg-Schur [45]. Ces méthodes ne sont pas utilisables lorsque la matrice $A$ est de grande taille ou creuse.

La deuxième catégorie est constituée des méthodes itératives pour les problèmes de grande taille. La plupart de ces méthodes sont les méthodes itératives de projections sur des sous-espaces de type Krylov, basées sur les algorithmes de type Arnoldi [36, 55, 68, 100, 102] pour construire une base orthonormée de l'espace de Krylov. Une caractéristique commune de toutes ces méthodes de projection est que les approximations $X_m$ de la solution exacte de l'équation de Lyapunov sont données sous forme factorisée, $X_m = Z_mZ_m^T$ avec $Z$ une matrice rang inférieur. De plus il y a la possibilité d'obtenir

une bonne approximation $X_m$ de rang inférieur (low rank approximate solution), il est soutenu par des résultats théoriques récents montrant la décroissance rapide des valeurs propres vers zéro de la solution exacte $X$, voir [61, 68, 82, 98, 102]. Cette approche est très importante lorsqu'il s'agit des problèmes de grande taille, puisque stocker la matrice $X_m$ complète en général nécessiterait une quantité significative de l'espace mémoire.

Nous proposons dans cette partie une nouvelle méthode pour résoudre l'équation de Lyapunov de grande taille déterminer l'approximation de la solution de l'équation de

## 2.2 Solution exacte de l'équation de Lyapunov

Dans cette section, nous allons donner l'expression exacte de la solution $X$ de l'équation de Lyapunov. Supposons que l'équation de Lyapnuov admette une solution unique $X$. Il a été montré dans [54, 105] que l'expression de la solution est la suivante

$$X = \sum_{j=1}^{p}\sum_{i=1}^{p} \gamma_{i,j} A^{i-1} B B^T (A^T)^{j-1},$$

où $p$ le degré du polynôme minimal de $A$ pour la matrice $B$ ($P(A)B = 0$) et la matrice $\mathcal{T} = (\gamma_{i,j})_{1\leq i,j\leq p}$ est la solution d'une équation matricielle de Lyapnuov de petite taille (Voir [103]).

Jbilou et Riquet ont proposé, dans [68], une méthode permettant de calculer les coefficients $\gamma_{i,j}$ à partir d'une base $F$-orthonormée de l'espace de Krylov

$$\mathcal{K}_p(A,\ B) = span\{B, AB, A^2B, ..., A^{p-1}B\},$$

construit par l'algorithme d'Arnoldi global. Ils ont aussi montré que la solution $X$ est de la forme

$$X = \mathcal{V}_p(\mathcal{T} \otimes I_s)\mathcal{V}_p^T, \tag{2.4}$$

où $\mathcal{V}_p$ est la matrice F-orthogonale construite par l'algorithme d'Arnoldi global associé $\mathcal{K}_p(A,\ B)$.

Comme la solution exacte est donnée par l'expression (2.4), l'approximation $X_m$ de la solution exacte est définie par :

$$X = \mathcal{V}_m(\mathcal{T} \otimes I_s)\mathcal{V}_m^T. \tag{2.5}$$

## 2.3 Méthode de Galerkin

Dans cette section, nous présentons une méthode de projection pour résoudre l'équation de Lyapunov de grande taille. En utilisant des projections dans des sous espaces de Krylov étendus (extended) globale avec la condition d'orthogonalité de Galerkin.

Nous allons chercher des approximations $X_m$ de la solution exacte $X$ de l'équation de Lyapunov sous la forme

$$X_m = \mathcal{V}_m(\mathcal{Y}_m^{GA} \otimes I_s)\mathcal{V}_m^T, \tag{2.6}$$

où $\mathcal{V}_m$ est la matrice F-orthogonale construite en appliquant $m$ itérations de l'algorithme d'Arnoldi étendu global à $\mathcal{K}_m^e(A,\ B)$. Donc on a les relations suivantes

$$\mathbb{T}_m = \mathcal{V}_m^T \diamond (A\mathcal{V}_m).$$

La condition d'orthogonalité de Petrov-Galerkin permet de donner l'équation de Lyapunov projetée suivante :

$$\mathbb{T}_m Y_m^{GA} + Y_m^{GA}\mathbb{T}_m^T + \|B\|_F^2 e_1^{2m}(e_1^{2m})^T = 0. \tag{2.7}$$

La solution de l'équation projeté au-dessus peut être obtenu par une méthode directe comme la méthode Hessenberg-Schur [45].

Pour réduire le calcul dans le test d'arrêter les itérations de la méthode proposée, nous donnons une borne supérieure (approximation supérieure) de la norme de Frobenius du résidu $\mathcal{R}(X_m)$.

**Théorème 2.3.1.** *Soient $Y_m$ la solution exacte de l'équation de Lyapunov projetée (2.7) et $X_m = \mathcal{V}_m(\mathcal{Y}_m^{GA}\otimes I_s)\mathcal{V}_m^T$ la solution approchée de l'équation de Lyapunov 2.2. La norme de Frobenius vérifie l'inégalité*

$$\| \mathcal{R}(X_m) \|_F \leq \sqrt{2(m+1)}\alpha := r_m^{GA}, \tag{2.8}$$

*où $\alpha = \| T_{m+1,m}Y_m \|_F$*

*Démonstration.* A l'étape $m$, nous avons :

$$\begin{aligned}\mathcal{R}(X_m) &= A\mathcal{V}_m(\mathcal{Y}_m^{GA} \otimes I_s)\mathcal{V}_m^T + \mathcal{V}_m(\mathcal{Y}_m^{GA} \otimes I_s)\mathcal{V}_m^T A^T + BB^T \\ &= \mathcal{V}_{m+1}(\overline{\mathbb{T}}_m \otimes I_s)(\mathcal{Y}_m^{GA} \otimes I_s)\mathcal{V}_m^T + \mathcal{V}_m(\mathcal{Y}_m^{GA} \otimes I_s)(\overline{\mathbb{T}}_m^T \otimes I_s)\mathcal{V}_{m+1}^T + \| B \|_F^2\ V_1V_1^T.\end{aligned}$$

En utilisant la relation

$$V_1 = \mathcal{V}_{m+1}[e_1^{(2m+2)} \otimes I_s],$$

et le fait que

$$\begin{aligned}
\mathcal{V}_m &= [V_1, V_2, V_3, ..., V_m] && (2.9)\\
&= [V_1, V_2, V_3, ..., V_{m+1}] \begin{pmatrix} I_{2s} & & & \\ & I_{2s} & & \\ & & \ddots & \\ & & & I_{2s} \\ & & & 0_{2s} \end{pmatrix} && (2.10)\\
&= [V_1, V_2, V_3, ..., V_{m+1}] \begin{pmatrix} [I_2 \otimes I_s] & & & \\ & [I_2 \otimes I_s] & & \\ & & \ddots & \\ & & & [I_2 \otimes I_s] \\ & & & [0_2 \otimes I_s] \end{pmatrix} && (2.11)\\
&= [V_1, V_2, V_3, ..., V_{m+1}] \left[ \begin{pmatrix} I_{2m} \\ 0_{2\times 2m} \end{pmatrix} \otimes I_s \right] && (2.12)\\
&= \mathcal{V}_{m+1}(\overline{I}_m \otimes I_s), \quad o\grave{u} \;\; \overline{I}_m = \begin{pmatrix} I_{2m} \\ 0_{2\times 2m} \end{pmatrix}. && (2.13)
\end{aligned}$$

Nous obtenons,

$$\begin{aligned}
\mathcal{R}(X_m) &= \mathcal{V}_{m+1} \Big[ (\overline{\mathbb{T}}_m \mathcal{Y}_m^{GA} \otimes I_s)(\overline{I}_m \otimes I_s) + (\overline{I}_m \otimes I_s)(\mathcal{Y}_m^{GA} \overline{\mathbb{T}}_m^T \otimes I_s) \\
&\quad + (\parallel B \parallel_F^2 e_1^{(2m+2)} (e_1^{(2m+2)})^T \otimes I_s) \Big] \mathcal{V}_{m+1}^T. \\
&= \mathcal{V}_{m+1} \Big[ \Big[ \overline{\mathbb{T}}_m \mathcal{Y}_m^{GA} \overline{I}_m + \overline{I}_m \mathcal{Y}_m^{GA} \overline{\mathbb{T}}_m^T + \parallel B \parallel_F^2 e_1^{(2m+2)} e_1^{(2m+2)^T} \Big] \otimes I_s \Big] \mathcal{V}_{m+1}^T. \\
&= \mathcal{V}_{m+1} \Bigg( \Bigg[ \begin{bmatrix} \mathbb{T}_m \\ T_{m+1,m} \end{bmatrix} \mathcal{Y}_m^{GA} \begin{bmatrix} I_{2m} \\ 0_{2\times 2m} \end{bmatrix}^T + \begin{bmatrix} I_{2m} \\ 0_{2\times 2m} \end{bmatrix} \mathcal{Y}_m^{GA} \begin{bmatrix} \mathbb{T}_m \\ T_{m+1,m} \end{bmatrix}^T \\
&\quad + \parallel B \parallel_F^2 \begin{bmatrix} e_1^{(2m)} \\ 0_{2\times 1} \end{bmatrix} \begin{bmatrix} e_1^{(2m)} \\ 0_{2\times 1} \end{bmatrix}^T \Bigg] \otimes I_s \Bigg) \mathcal{V}_{m+1}^T. \\
&= \mathcal{V}_{m+1} \left[ \begin{pmatrix} \mathbb{T}_m Y_m + Y_m \mathbb{T}_m^T + \parallel B \parallel_F^2 e_1^{(2m)} e_1^{(2m)^T} & Y_m T_{m+1,m}^T \\ T_{m+1,m} Y_m & 0 \end{pmatrix} \otimes I_s \right] \mathcal{V}_{m+1}^T.
\end{aligned}$$

Comme $Y$ solution de l'équation $\mathbb{T}_m Y_m + Y_m \mathbb{T}_m^T + \parallel B \parallel_F^2 e_1^{(2m)} e_1^{(2m)^T} = 0$,

$$\mathcal{R}_m(X_m) = \mathcal{V}_{m+1} \left[ \begin{pmatrix} 0 & Y_m T_{m+1,m}^T \\ T_{m+1,m} Y_m & 0 \end{pmatrix} \otimes I_s \right] \mathcal{V}_{m+1}^T.$$

Par conséquent

$$\|\mathcal{R}_m(X_m)\|_F \leq \left\|\mathcal{V}_{m+1}^T\right\|_F \times \left\|\mathcal{V}_{m+1}\left[\begin{pmatrix} 0 & Y_m E_m T_{m+1,m}^T \\ T_{m+1,m}E_m^T Y_m & 0 \end{pmatrix} \otimes I_s\right]\right\|_F$$

Comme $\{V_1, V_2, ..., V_{m+1}\}$ est une base $F$-orthonormée de l'espace de Krylov étendu $\mathcal{K}_{m+1}(A, B)$ et $\mathcal{V}_{m+1} = [V_1, V_2, ..., V_{m+1}]$, nous avons $\|\mathcal{V}_{m+1}\|_F = \sqrt{m+1}$. D'autre part

$$\begin{aligned}\left\|\mathcal{V}_{m+1}\left[\begin{pmatrix} 0 & Y_m e_1^m T_{m+1,m}^T \\ T_{m+1,m}e_m^T Y_m & 0 \end{pmatrix} \otimes I_s\right]\right\|_F &= \left\|\begin{pmatrix} 0 & Y_m E_m T_{m+1,m}^T \\ T_{m+1,m}(e_1^{2m})^T Y_m & 0 \end{pmatrix}\right\|_F \\ &= \sqrt{2}\left\|T_{m+1,m}E_m^T Y_m\right\|_F\end{aligned}$$

Par conséquent

$$\|\mathcal{R}_m(X_m)\|_F \leq \sqrt{2(m+1)}\alpha,$$

où $\alpha = \left\|T_{m+1,m}E_m^T Y_m\right\|_F$. □

Pour économiser de la mémoire il est possible d'écrire $X_m$ sous forme factorisée. Soit la décomposition en valeurs singulières de la matrice $Y_m = P\Sigma P^T$, où $\Sigma = diag(\sigma_1, \sigma_2, ..., \sigma_{2m})$ et $\sigma_1 \geq \sigma_2 \geq ... \geq \sigma_{2m}$. Nous fixons *dtol* et définissons $P_l$ la matrice constituée des k premières colonnes de $l$ correspondant aux $l$ valeurs singulières supérieures ou égales à *dtol*.

En posant $\Sigma_l = diag(\sigma_1, \sigma_2, ..., \sigma_l)$ et $Z_m = \mathcal{V}_m P_l(\Sigma_l^{1/2} \otimes I_s)$, nous obtenons l'approximation $X_m = Z_m Z_m^T$ (qui est la meilleure approximation de rang $l$ de la matrice $X_m$, on pourra se référer à [44, 56] pour un exposé complet sur ce sujet).

L'algorithme de la méthode d'Arnoldi étendu global pour résoudre l'équation de Lyapunov, se résume comme suit :

---

**Algorithme 2.1 : pour résoudre l'équation de Lyapunov (GA-Lyap)**

---

1. On choisit une tolérance $\epsilon > 0$ et un nombre maximal d'itérations $m_{max}$
2. **Pour** $m = 1, 2, 3, ..., m_{max}$
3. Construire les matrices $\mathcal{V}_m$ et $\mathbb{T}_m$ par l'algorithme d'Arnoldi étendu global appliqué au couple $(A, B)$.
4. Résoudre l'équation de Lyapunov projetée suivante

$$\mathbb{T}_m Y_m^{GA} + Y_m^{GA}\mathbb{T}_m^T + \|B\|_F^2 e_1^{2m}(e_1^{2m})^T = 0.$$

5. Calculer la norme de Frobenius de $r_m^{GA}$ en utilisant (2.8).
   - **Si** $\|\mathcal{R}_m\| < \epsilon$ , **Aller à 8**
   - **Sinon** $m = m + 1$

– **Fin si**

6. **Fin pour** $m$
7. Calculer la décomposition en valeurs singulières (SVD) de la matrice $Y_m^{GA}$, c'est à dire $Y_m^{GA} = \mathbb{V}\Sigma\mathbb{V}^T$, où $\Sigma = diag(\sigma_1, \sigma_2, ..., \sigma_{2mr})$ et $\sigma_1 \geq \sigma_2 \geq ... \geq \sigma_{2mr}$ ; trouver $l$ telle que $\sigma_{l+1} \leq tol_{trn} < \sigma_l$ et soit $\Sigma_l = diag(\sigma_1, \sigma_2, ..., \sigma_l)$ ; $Z_m = \mathcal{V}_m \mathbb{V}_l (\Sigma_l^{1/2} \otimes I_s)$.
8. La solution approchée $X_m$ est donnée par $X_m = Z_m\, Z_m^T$.

---

## 2.4 Méthode de minimisation du résidu

Dans ce paragraphe, nous proposons une nouvelle méthode de projection pour résoudre l'équation de Lyapunov de grande taille. En utilisant des projections sur des sous espaces de Krylov étendus (extended) globale avec la condition de minimisation de la norme de Frobenius du résidu associé à l'équation de Lyapunov.

Les approximations $X_m$ de la solution exacte $X$ de l'équation de Lyapunov sont de la forme

$$X_m = \mathbb{V}_m(\mathcal{Y}_m \otimes I_s)\mathbb{V}_m^T, \tag{2.14}$$

où $\mathbb{V}_m$ est la matrice F-orthogonale construite en appliquant $m$ itérations de l'algorithme d'Arnoldi étendu global à $\mathcal{K}_e^g(A,\ B)$.
Nous allons chercher des approximations $X_m^{MR}$ qui minimisent la norme du résidu $\mathcal{R}(X_m) = AX_m + X_m A^T + BB^T$, où $X_m = \mathbb{V}_m(\mathcal{Y}_m \otimes I_s)\mathbb{V}_m^T$, c'est-à dire :

$$X_m^{MR} = \arg \min_{X_m = \mathbb{V}_m(\mathcal{Y}_m \otimes I_s)\mathbb{V}_m^T} \|AX_m + X_m A^T + BB^T\| \tag{2.15}$$

Comme $X_m^{MR} = \mathbb{V}_m(\mathcal{Y}_m^{MR} \otimes I_s)\mathbb{V}_m^T$, nous avons le résultat suivant

**Théorème 2.4.1.** *Soit $\mathbb{V}_m$ la matrice F-orthogonale construite en appliquant $m$ itérations de l'algorithme d'Arnoldi étendu globale à $(A, B)$. Alors le problème de minimisation :*

$$X_m^{MR} = \arg \min_{X_m = \mathbb{V}_m(\mathcal{Y}_m \otimes I_s)\mathbb{V}_m^T} \left\| AX_m + X_m A^T + BB^T \right\|_F$$

*peut s'écrire sous la forme*

$$\mathcal{Y}_m^{MR} = \arg \min_{\mathcal{Y}_m} \left\| \overline{\mathbb{T}}_m \mathcal{Y}_m \begin{bmatrix} I_{2m} \\ 0_{2\times 2m} \end{bmatrix}^T + \begin{bmatrix} I_{2m} \\ 0_{2\times 2m} \end{bmatrix} \mathcal{Y}_m \overline{\mathbb{T}}_m^T + \begin{bmatrix} \|B\|_F^2\, e_1^{(2m)} e_1^{(2m)^T} & 0 \\ 0 & 0 \end{bmatrix} \right\|_F . \tag{2.16}$$

*Démonstration.* A l'étape $m$, nous avons :

$$\begin{aligned}
&\min_{X_m=\mathbb{V}_m(\mathcal{Y}_m\otimes I_s)\mathbb{V}_m^T} \left\| AX_m + X_m A^T + BB^T \right\|_F \\
&= \min_{\mathcal{Y}_m} \left\| A\mathcal{V}_m \left(\mathcal{Y}_m \otimes I_s\right) \mathcal{V}_m^T + \mathcal{V}_m \left(\mathcal{Y}_m \otimes I_s\right) \mathcal{V}_m^T A^T + \|B\|_F^2 V_1 V_1^T \right\|_F \\
&= \min_{\mathcal{Y}_m} \left\| \mathcal{V}_{m+1} \left[ \left(\overline{\mathbb{T}}_m \mathcal{Y}_m \otimes I_s\right) \left( \begin{bmatrix} I_{2m} \\ 0_{2\times 2m} \end{bmatrix}^T \otimes I_s \right) + \left( \begin{bmatrix} I_{2m} \\ 0_{2\times 2m} \end{bmatrix} \otimes I_s \right) \left( \mathcal{Y}_m \overline{\mathbb{T}}_m^T \otimes I_s \right) \right. \right. \\
&\left. \left. + \left( \|B\|_F^2 \, e_1^{(2m+2)} e_1^{(2m+2)^T} \right) \otimes I_s \right] \mathcal{V}_{m+1}^T \right\|_F . \\
&= \min_{\mathcal{Y}_m} \left\| \mathcal{V}_{m+1} \left[ \left( \overline{\mathbb{T}}_m \mathcal{Y}_m \begin{bmatrix} I_{2m} \\ 0_{2\times 2m} \end{bmatrix}^T + \begin{bmatrix} I_{2m} \\ 0_{2\times 2m} \end{bmatrix} \mathcal{Y}_m \overline{\mathbb{T}}_m^T + \|B\|_F^2 \begin{bmatrix} e_1^{(2m)} \\ 0_{2\times 1} \end{bmatrix} \begin{bmatrix} e_1^{(2m)} \\ 0_{2\times 1} \end{bmatrix}^T \right) \otimes I_s \right] \right.
\end{aligned}$$

Comme $\mathcal{V}_{m+1}$ independent de $\mathcal{Y}_m$ donc nous avons :

$$\mathcal{Y}_m^{MR} = \arg\min_{\mathcal{Y}_m} \left\| \overline{\mathbb{T}}_m \mathcal{Y}_m \begin{bmatrix} I_{2m} \\ 0_{2\times 2m} \end{bmatrix}^T + \begin{bmatrix} I_{2m} \\ 0_{2\times 2m} \end{bmatrix} \mathcal{Y}_m \overline{\mathbb{T}}_m^T + \begin{bmatrix} \|B\|_F^2 \, e_1^{(2m)} e_1^{(2m)^T} & 0 \\ 0 & 0 \end{bmatrix} \right\|_F .$$

□

Pour résoudre le problème de minimisation projeté (2.16), il existe plusieurs stratégies possibles, nous proposons dans la section suivante deux méthodes itératives pour déterminer la solution approchée de (2.16) .

## 2.5 Méthodes itératives pour résoudre le problème réduit

### 2.5.1 LSQR global

Dans cette section, nous proposons une méthode itérative de type LSQR global (GL-LSQR) pour résoudre le problème de minimisation projeté (2.16). La méthode LSQR classique proposée par Paige et Saunders (1982) est une méthode itérative pour résoudre les systèmes linéaires ou les problèmes aux moindres carrés. Pour plus de précision voir [95]. L'algorithme LSQR est basé sur l'algorithme de Golub-Kahan [43], en utilisant la procédure de bidiagonalisation de Lanczos dans [95]. La version globale de cette méthode est donnée dans [106], il y a aussi la version bloc voir [75]. Dans ce qui suit, nous allons proposer une méthode globale-LSQR (GLSQR) pour résoudre les problèmes de minimisation de type $\min_Y \|\mathcal{L}_m(Y) - \mathcal{C}\|_F$, où $\mathcal{L}_m$ est un opérateur linéaire. Comme notre objectif est de résoudre le problème de minimisation suivant :

$$\mathcal{Y}_m^{MR} = \arg\min_{\mathcal{Y}_m} \left\| \overline{\mathbb{T}}_m \mathcal{Y}_m \begin{bmatrix} I_{2m} \\ 0_{2\times 2m} \end{bmatrix}^T + \begin{bmatrix} I_{2m} \\ 0_{2\times 2m} \end{bmatrix} \mathcal{Y}_m \overline{\mathbb{T}}_m^T + \begin{bmatrix} \|B\|_F^2 \, e_1^{(2m)} e_1^{(2m)^T} & 0 \\ 0 & 0 \end{bmatrix} \right\|_F ,$$

Nous proposons le problème sous la forme

$$\min_{Y} \|\mathcal{L}_m(\mathcal{Y}) - \mathcal{C}\|_F, \tag{2.17}$$

où

$$\mathcal{L}_m(\mathcal{Y}) = \bar{\mathbb{T}}_m \mathcal{Y} \begin{pmatrix} I & 0 \end{pmatrix} + \begin{pmatrix} I \\ 0 \end{pmatrix} \mathcal{Y} \bar{\mathbb{T}}_m^T \tag{2.18}$$

et

$$\mathcal{C} = - \begin{pmatrix} \|B\|_F^2 \, e_1^{(2m)} e_1^{(2m)^T} & 0 \\ 0 & 0 \end{pmatrix}.$$

L'opérateur adjoint de $\mathcal{L}_m$ par rapport au produit scalaire de Frobenius est donné par

$$\mathcal{L}_m{}^T(Y) = \bar{\mathbb{T}}_m^T Y \begin{pmatrix} I \\ 0 \end{pmatrix} + \begin{pmatrix} I & 0 \end{pmatrix} Y \bar{\mathbb{T}}_m^B. \tag{2.19}$$

**Procédure de bidiagonalisation globale de Lanczos (Golub-Kahan)** :
Cette procédure itérative est initialisée par

$$\begin{array}{ll} \beta_1 \widetilde{U}_1 = \mathcal{C} & \beta_1 = \|\mathcal{C}\|_F \\ \alpha_1 V_1 = \mathcal{L}_m{}^T\left(\widetilde{U}_1\right) & \alpha_1 = \left\|\mathcal{L}_m{}^T\left(\widetilde{U}_1\right)\right\|_F. \end{array}$$

et les relations de récurrence,

$$\begin{cases} \beta_{i+1}\widetilde{U}_{i+1} = \mathcal{L}_m(\widetilde{V}_i) - \alpha_i \widetilde{U}_i \\ \alpha_{i+1}\widetilde{V}_{i+1} = \mathcal{L}_m{}^T\left(\widetilde{U}_i\right) - \beta_i \widetilde{V}_i \end{cases}$$

les scalaires $\alpha_i > 0$ et $\beta_i > 0$ sont choisi de telle sorte que $\left\|\widetilde{U}_i\right\|_F = \left\|\widetilde{V}_i\right\|_F = 1$, pour tout $i = 1, 2, \ldots$.
On définit les matrices suivantes

$$\widetilde{\mathbb{U}}_k := [\widetilde{U}_1, \widetilde{U}_2, ..., \widetilde{U}_k];$$

$$\widetilde{\mathbb{V}}_k := [\widetilde{V}_1, \widetilde{V}_2, ..., \widetilde{V}_k]$$

et

$$\bar{T}_k := \begin{bmatrix} \alpha_1 & & & \\ \beta_2 & \alpha_2 & & \\ & \beta_3 & \ddots & \\ & & \ddots & \alpha_k \\ & & & \beta_{k+1} \end{bmatrix}.$$

Par construction les matrices $\widetilde{U}_i$ et $\widetilde{V}_i$ sont $F$-orthonormées, ce qui signifie qu'ils sont orthonormées par rapport au produit scalaire de Frobenius : $\langle U_i, U_j\rangle_F = \delta_{i,j}$ et $\langle V_i, V_j\rangle_F = \delta_{i,j}$ où $\delta_{i,j}$ est le symbole de Kronecker. L'utilisation de la procédure de bidiagonalisation globale de Lanczos permet de trouver à chaque itération $k$ des approximations $Y^k$ de la solution exacte $Y_m^{MR}$ du problème de minimisation réduit (2.16). Nous pouvons écrire les relations de récurrence précédentes sous forme matricielle comme suit

$$[\mathcal{L}_m(\widetilde{V}_1), \mathcal{L}_m(\widetilde{V}_2), ..., \mathcal{L}_m(\widetilde{V}_k)] = \widetilde{\mathbb{U}}_{k+1}(\bar{T}_k \otimes I_s). \tag{2.20}$$

La méthode consiste à chercher la solution approchée $Y^k$ ayant la forme suivante

$$Y^k = \sum_{i=1}^{k} z^{(i)} \widetilde{V}_i.$$

Nous appliquons l'opérateur $\mathcal{L}_m$ à $Y^k$, nous obtenons

$$\begin{aligned}
\mathcal{L}_m(Y^k) &= \mathcal{L}_m\left(\sum_{i=1}^{k} z^{(i)}\widetilde{V}_i\right) \\
&= \sum_{i=1}^{k} z^{(i)}\mathcal{L}_m(\widetilde{V}_i) \\
&= [\mathcal{L}_m(\widetilde{V}_1), ..., \mathcal{L}_m(\widetilde{V}_k)]\,(z_k \otimes I_s)\,, \quad \text{où} \quad z_k = (z^{(1)}, z^{(2)}, \cdots, z^{(k)})^T \\
&= \widetilde{\mathbb{U}}_{k+1}(\bar{T}_k \otimes I_s)(z_k \otimes I_s) \\
&= \widetilde{\mathbb{U}}_{k+1}(\bar{T}_k z_k \otimes I_s).
\end{aligned}$$

et

$$\begin{aligned}
\left\|\mathcal{C} - \mathcal{L}_m(Y^k)\right\|_F &= \left\|\beta_1\widetilde{U}_1 - \widetilde{\mathbb{U}}_{k+1}(\bar{T}_k z_k \otimes I_s)\right\|_F \\
&= \left\|\widetilde{\mathbb{U}}_{k+1}(\beta_1 e_1 \otimes I_s) - \widetilde{\mathbb{U}}_{k+1}(\bar{T}_k z_k \otimes I_s)\right\|_F \\
&= \left\|\mathbb{U}_{k+1}\left[(\beta_1 e_1 \otimes I_s) - (\bar{T}_k z_k \otimes I_s)\right]\right\|_F \\
&= \left\|(\beta_1 e_1 - \bar{T}_k z_k) \otimes I_s\right\| = \|\beta_1 e_1 - T_k z_k\|_F\,.
\end{aligned}$$

Donc le problème `minimize` $\left\|\mathcal{C} - \mathcal{L}_m(Y^k)\right\|_F$ est alors équivalent à résoudre le problème suivant

$$\texttt{minimize} \quad \left\|\beta_1 e_1 - \bar{T}_k z_k\right\|_2. \tag{2.21}$$

Alors, la résolution du problème de minimisation (2.16), par la méthode LSQR globale (Gl-LSQR) permet de donner une solution approchée de la solution exacte $Y_m^{MR}$ du problème (2.16).

La méthode LSQR globale (Gl-LSQR) est résumé dans l'algorithme suivant :

**Algorithme 2.2 : Algorithme LSQR globale (Gl-LSQR)**

1. Initialisation : $X_0 = 0$
   $\beta_1 = \|N\|_F$, $\widetilde{U}_1 = N/\beta_1$,
   $\alpha_1 = \left\|(\mathcal{L}_m)^T(\widetilde{U}_1)\right\|_F$,
   $\widetilde{V}_1 = (\mathcal{L}_m)^T(\widetilde{U}_1)/\alpha_1$,
   $\widetilde{W}_1 = \widetilde{V}_1$, $\overline{\Phi_1} = \beta_1$, $\overline{\rho_1} = \alpha_1$.
2. Pour $i = 1, 2, ..., kmax$ :
   $\widetilde{W}_i = \mathcal{L}_m(\widetilde{V}_i) - \alpha_i\widetilde{U}_i$, $\beta_{i+1} = \left\|\widetilde{W}_i\right\|$,
   $\widetilde{U}_{i+1} = \widetilde{W}_i/\beta_{i+1}$,
   $L_i = (\mathcal{L}_m)^T(\widetilde{U}_{i+1}) - \beta_{i+1}\widetilde{V}_i$,
   $\alpha_{i+1} = \|L_i\|_F$,
   $\widetilde{V}_{i+1} = L_i/\alpha_{i+1}$,
   $\rho_i = \sqrt{(\overline{\rho_1}^2 + \beta_{i+1}^2)}$
   $c_i = \overline{\rho_1}/\rho_1$
   $s_i = \beta_{i+1}/\rho_1$
   $\theta_{i+1} = s_i\alpha_{i+1}$
   $\overline{\rho}_{i+1} = c_i\alpha_{i+1}$
   $\Phi_i = c_i\overline{\Phi_i}$
   $\overline{\Phi}_{i+1} = s_i\Phi_i$
   $Y^i = Y^{i-1} + (\Phi_i/\rho_i)\widetilde{W}_i$
   $W_{i+1} = V_{i+1} - (\theta_{i+1}/\rho_i)\widetilde{W}_i$.
   Si $\left|\overline{\Phi}_{i+1}\right|$ est petit alors on s'arrête.
3. Fin pour.

L'algorithme précédent permet de calculer une solution approchée $Y^k$ avec $1 \leq k \leq kmax$, du problème de minimisation (2.20). Afin de savoir si une solution $X_m^{MR}$ approche suffisamment bien la solution exacte $X$, nous devons être en mesure de calculer de façon simple et peu couteuse le résidu $\mathcal{R}_m$ associé à $X_m^{MR}$. Ensuite, nous donnons l'expression de $\|\mathcal{R}_m(X_m)\|$ selon la norme du résidu associé à l'algorithme LSQR globale (Gl-LSQR) appliquée au problème de minimisation réduit.

**Théorème 2.5.1.** *Soit $X_m^{MR} = \mathbb{V}_m(\mathcal{Y}_m^{MR} \otimes I_s)\mathbb{V}_m^T$ la solution approchée de l'équation matricielle de Lyapunov obtenu après m itérations de l'algorithme d'Arnoldi étendu global et $\mathcal{Y}_m^{MR}$ la solution exacte du problème*

$$\mathcal{Y}_m^{MR} = \arg\min_{\mathcal{Y}_m} \left\| \overline{\mathbb{T}}_m\mathcal{Y}_m \begin{bmatrix} I_{2m} \\ 0_{2\times 2m} \end{bmatrix}^T + \begin{bmatrix} I_{2m} \\ 0_{2\times 2m} \end{bmatrix} \mathcal{Y}_m\overline{\mathbb{T}}_m^T + \begin{bmatrix} \|B\|_F^2\, e_1^{(2m)}e_1^{(2m)^T} & 0 \\ 0 & 0 \end{bmatrix} \right\|_F.$$

*Alors, nous avons :*

$$\|\mathcal{R}_m(X_m^{MR})\|_F \leq \sqrt{m+1}\Phi$$

*où* $\Phi = \mid \overline{\Phi}_{kmax} \mid$ *est donné dans l'algorithme 1 (Gl-LSQR).*

*Démonstration.* nous avons

$$\begin{aligned}
\|\mathcal{R}(X_m^{MR})\|_F &= \|AX_m^{MR} + X_m^{MR}A^T + BB^T\|_F, \quad \text{où} \quad X_m^{MR} = \mathbb{V}_m(\mathcal{Y}_m^{MR} \otimes I_s)\mathbb{V}_m^T \\
&= \left\|\mathbb{V}_{m+1}\left[\left(\mathcal{L}_m(\mathcal{Y}_m^{MR}) - \mathcal{C}\right) \otimes I_s\right]\mathbb{V}_{m+1}^T\right\|_F \\
&\leq \|\mathbb{V}_{m+1}\|_F \times \left\|\left[\left(\mathcal{L}_m(\mathcal{Y}_m^{MR}) - \mathcal{C}\right) \otimes I_s\right]\mathbb{V}_{m+1}^T\right\|_F \\
&\leq \sqrt{m+1}\left\|\mathcal{L}_m(\mathcal{Y}_m^{MR}) - \mathcal{C}\right\|_F \\
&\leq \sqrt{m+1}\min_{\mathcal{Y}} \|\mathcal{L}_m(\mathcal{Y}) - \mathcal{C}\|_F \\
&\leq \sqrt{m+1}\Phi.
\end{aligned}$$

□

### 2.5.2 La méthode du gradient conjugué globale préconditionné

Dans ce paragraphe, nous considérons la méthode du gradient conjugué globale préconditionnés (PGCG) pour résoudre le problème de minimisation réduit (2.16). La méthode du gradient conjugué préconditionné (PCG) classique appliquée à l'équation normale. L'équation normale associée à (2.16) est donnée par

$$\mathcal{L}_m^*(\mathcal{L}_m Y)) = \mathcal{L}_m^*(\mathcal{C}), \tag{2.22}$$

où $\mathcal{L}_m$ et $\mathcal{L}_m^*$ les opérateurs donnés par (2.18) et (2.19) respectivement.

En conséquence,

$$\min_Y \|\mathcal{L}_m(Y) - \mathcal{C}\|_F \iff \mathcal{L}_m^*(\mathcal{L}_m(Y) - \mathcal{C}) = 0.$$

La matrice $\overline{\mathbb{T}}_m$ sous la forme

$$\overline{\mathbb{T}}_m = \begin{pmatrix} \mathbb{T}_m \\ h_m \end{pmatrix},$$

où $h_m$ représentent les $2r$ dernières lignes de la matrice $\overline{\mathbb{T}}_m$. Par conséquent, l'équation normale (2.22) peut se écrire comme suit

$$\overline{\mathbb{T}}_m^T\overline{\mathbb{T}}_m Y + Y\overline{\mathbb{T}}_m^T\overline{\mathbb{T}}_m + \mathbb{T}_m^T Y \mathbb{T}_m^T + \mathbb{T}_m Y \mathbb{T}_m - \mathcal{C}_1 = 0, \tag{2.23}$$

où $\mathcal{C}_1 = \mathcal{L}_m^T(\mathcal{C})$. Considérons la décomposition en valeurs singulières (SVD) de $\overline{\mathbb{T}}_m$ :

$$\overline{\mathbb{T}}_m = \overline{U}\,\overline{\Sigma}\,\overline{U}^T, \tag{2.24}$$

alors nous obtenons la décomposition suivante

$$\overline{\mathbb{T}}_m^T \overline{\mathbb{T}}_m = Q D Q^T, \tag{2.25}$$

où $Q = \overline{U}$, et $D = \overline{\Sigma}^T \overline{\Sigma}$.

Soient $\widetilde{Y} = Q^T Y Q$ et $\widetilde{\mathcal{C}} = Q^T \mathcal{C}_1 Q$, donc l'équation normale (2.22) peut aussi s'écrire

$$D\widetilde{Y} + \widetilde{Y} D + \widetilde{\mathbb{T}}_m \widetilde{Y} \widetilde{\mathbb{T}}_m + \widetilde{\mathbb{T}}_m^T \widetilde{Y} \widetilde{\mathbb{T}}_m^T - \widetilde{\mathcal{C}} = 0, \tag{2.26}$$

où $\widetilde{\mathbb{T}}_m = Q^T \mathbb{T}_m Q$ et $\widetilde{Y} = Q^T Y Q$. Cette expression propose que l'on peut utiliser l'opérateur matriciel suivant comme un préconditionneur

$$\mathfrak{M}(\widetilde{Y}) = D_A \widetilde{Y} + \widetilde{Y} D_B. \tag{2.27}$$

On peut voir que l'expression (2.22) correspondant à l'équation normale de l'opérateur matriciel suivant

$$\widetilde{\mathcal{L}}_m(\widetilde{Y}) = \widetilde{\mathcal{T}}_m \, \widetilde{Y} \begin{pmatrix} Q^T & 0 \end{pmatrix} + \begin{pmatrix} Q \\ 0 \end{pmatrix} \widetilde{Y} \, \widetilde{\mathcal{T}}_m^T \tag{2.28}$$

où $\widetilde{\mathcal{T}}_m = \overline{\mathbb{T}}_m \, Q$. Par conséquent, l'algorithme du gradient conjugué préconditionné globale est obtenu en appliquant le préconditionneur (2.27) à l'équation normale associée à l'opérateur matriciel défini par (2.28). Ceci est résumé dans l'algorithme suivant :

**Algorithme 2.3 : Algorithme du gradient conjugué globale préconditionné(PGCG)**

1. On choisit une tolérance $tol > 0$, et un nombre maximal d'itérations $j_{max}$ ;
   Choisir $\widetilde{Y}_0$ et on fixe $\widetilde{R}_0 = \mathcal{C} - \widetilde{\mathcal{L}}_m(\widetilde{Y}_0)$ ; $S_0 = \widetilde{\mathcal{L}}_m^*(\widetilde{R}_0)$, $Z_0 = \mathfrak{M}^{-1}(S_0)$ ; $P_0 = S_0$.
2. **Pour** $j = 0, 1, 2, ..., j_{max}$
   (a) $W_j = \widetilde{\mathcal{L}}_m(P_j)$
   (b) $\alpha_j = \langle S_j, Z_j \rangle_F / |W_j|_F^2$
   (c) $\widetilde{Y}_{j+1} = \widetilde{Y}_j + \alpha_j P_j$
   (d) $\widetilde{R}_{j+1} = \widetilde{R}_j - \alpha_j W_j$
   (e) **Si** $\|\widetilde{R}_{j+1}\|_F < tol$. **Arrêt**
   **Sinon**
   (f) $S_{j+1} = \widetilde{\mathcal{L}}_m^*(\widetilde{R}_{j+1})$
   (g) $Z_{j+1} = \mathfrak{M}^{-1}(S_{j+1})$
   (h) $\beta_j = \langle S_{j+1}, Z_{j+1} \rangle_F / \langle S_j, Z_j \rangle_F$
   (i) $P_{j+1} = Z_{j+1} + \beta_j P_j$.
3. **Fin Pour**.

On note que l'utilisation du préconditionneur $\mathfrak{M}$ nécessite à chaque itération la résolution d'une équation de Sylvester. Comme la matrice $D$ de ces équations matricielles de Lyapunov est une matrice diagonale, ceci réduit les coûts de calcul.

Notons que le calcul de la norme du résidu $\mathcal{R}(X_m^{MR})$ à chaque itération $m$ nécessite seulement la connaissance de $Y_m^{MR}$ sans avoir à construire la solution approximative $X_m^{MR}$. Cette norme du résidu $\mathcal{R}(X_m^{MR})$ peut être calculée par

$$r_m := \mathcal{R}(X_m^{MR}) = \left\| \left[ \overline{\mathbb{T}}_m \mathcal{Y}_m \begin{bmatrix} I_{2m} \\ 0 \end{bmatrix}^T + \begin{bmatrix} I_{2m} \\ 0 \end{bmatrix} \mathcal{Y}_m \overline{\mathbb{T}}_m^T + \begin{bmatrix} \|B\|_F^2 \, e_1^{(2m)} e_1^{(2m)^T} & 0 \\ 0 & 0 \end{bmatrix} \right] \right\|_F . \tag{2.29}$$

La solution approximative $X_m^{MR}$ n'est calculée que si la convergence est réalisée. Dans la section suivante, nous proposons des méthodes (directes ou itératives) pour calculer la solution $Y_m^{MR}$ du problème de minimisation réduit (2.15).

Pour économiser la place de la mémoire, il est possible d'écrire $X_m$ sous forme factorisée. Soit $Y_m = P\Sigma P^T$, la décomposition en valeurs singulières de la matrice $Y_m$, où $\Sigma \in \mathbb{R}^{2m \times 2m}$ désigne la matrice des valeurs singulières de $Y_m$ rangées dans l'ordre décroissant, $P$ est une matrice unitaire. Nous fixons un seuil $dtol$ et définissons $P_k$ la matrice constituée des $k$ premières colonnes de $P$ correspondant aux $k$ valeurs singulières supérieures ou égales à $dtol$. Nous obtenons la décomposition en valeur singulière tronquée

$$Y_m \approx P_k \, \Sigma_k \, P_k^T$$

où $\Sigma_k = \text{diag}[\sigma_1, \ldots, \sigma_k]$. Soit $Z_m = \mathbb{V}_m P_k (\Sigma_k^{1/2} \otimes I_s)$, nous obtenons alors l'approximation de $X_m$ sous la forme d'une expression factorisée

$$X_m \approx Z_m Z_m^T. \tag{2.30}$$

Cette factorisation est très importante pour les problèmes de grande dimension, quand on n'a pas besoin de calculer l'approximation de la solution $X_m$, mais on a besoin de la stocker à chaque itération.

La méthode de minimisation du résidu MR pour résoudre l'équation de Lyapunov est résumé dans l'algorithme suivant

---

**Algorithme 2.4 : pour résoudre l'équation de Lyapunov (MR-LE)**

---

1. On choisit une tolérance $tol > 0$ et un nombre maximum d'itérations $iter_{max}$.
2. Pour $m = 1, 2, 3, ..., iter_{max}$
3. Construire les matrices $\mathbb{V}_m$ et $\overline{\mathbb{T}}_m$ par l'algorithme EGA appliqué à $(A,\ B)$.
4. Résoudre le problème de minimisation

$$\mathcal{Y}_m^{MR} = \arg\min_{\mathcal{Y}_m} \|\mathcal{L}_m(\mathcal{Y}_m) + C\|_F .$$

5. Si $\|\mathcal{R}_m\|_F \leq tol$, Arrêt,
6. La solution approchée $X_m$ est donnée par $X_m = Z_m Z_m^T$.

---

## 2.6 Exemples numériques

Dans cette section, nous présentons quelques exemples numériques de l'équation algébrique de Lyapunov de grande taille. Nous allons comparer l'approche de Galerkin (GA) avec notre approche de minimisation de résidu (MR). En utilisant : la méthode LSQR globale (Gl-LSQR) et la méthode du gradient conjugué globale préconditionné (PGCG) pour résoudre le problème de minimisation réduit. Les algorithmes ont été codés avec Matlab 2009. A chaque itération $m$ de l'algorithme d'Arnoldi étendu global, l'équation matricielle de Lyapunov projetée de taille $2m \times 2m$ a été résolue par la méthode directe de Bartels-Stewart [11]. Pour résoudre le problème de minimisation réduit, on s'arrête si la norme de Frobenius du résidu est inférieure à $tol_p = 10^{-10}$ ou quand le nombre maximum d'itérations $kmax = 500$ est atteint. Pour l'algorithme minimisation de résidu (MR) et l'algorithme de Galerkin on s'arrête lorsque la norme de Frobenius du résidu est inférieure à une certaine tolérance $tol = 10^-12$ ou bien le nombre maximum d'itérations

$itermax = 50$. Dans tous les tests, les coefficients de la matrice $B$ ont été générés par des valeurs aléatoires uniformément distribuées dans l'intervalle $[0,\ 1]$. La matrice $A$ est obtenue par la discrétisation par différences finies de l'opérateur suivant

$$L_u = \Delta u - f_1(x,y)\frac{\partial u}{\partial x} + f_2(x,y)\frac{\partial u}{\partial y} + g(x,y), \tag{2.31}$$

dans le domaine $[0,1] \times [0,1]$ avec les condition de Dirichlet homogène. La taille de la matrice construite est $n = n_0^2$, où $n_0$ est le nombre de points de la grille dans chaque direction. La discrétisation de l'opérateur $L_u$ donne des matrices qu'on pourra trouver dans la bibliothèque `Lyapack` [89] et désignés par

$$\text{A = fdm2d\_matrix(n0,'f\_1(x,y)','f\_2(x,y)','g(x,y)').}$$

Les figures qui apparaissent dans la suite, nous donnent les courbes représentant la norme de Frobenius du résidu dans une échelle logarithmique en fonction du nombre d'itérations $m$.

### 2.6.1 Exemple 1

Dans ce premier exemple, la matrice $A$ est donnée par :

$$A = fdm(n0, x^2 + 2y, e^{y+x}, 5).$$

Pour $r = 3$ et $n = 6400$, nous avons :

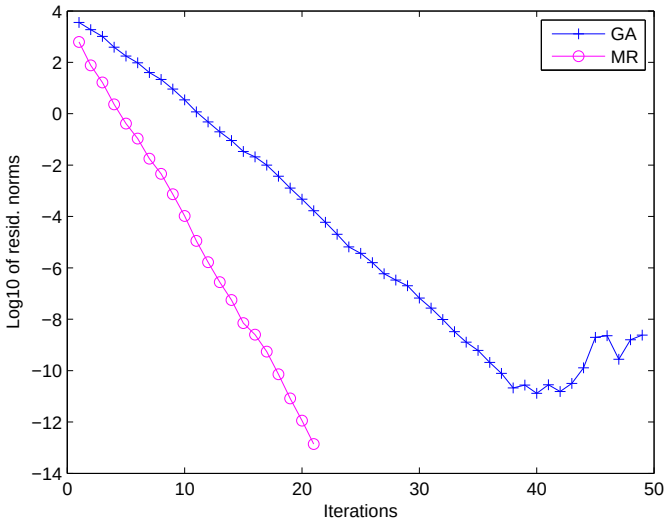


FIGURE 2.1: Résultats de l'exemple 1 : MR(PCGG)

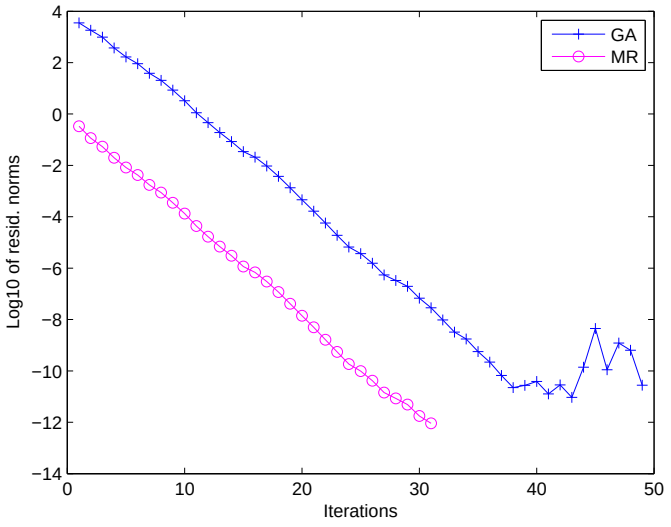


FIGURE 2.2: Résultats de l'exemple 1 : MR(GL-LSQR)

### 2.6.2 Exemple 2

Dans ce deuxième exemple, la matrice $A$ est donnée par :

$$A = fdm(n0, x^2 + y^2, sin(y + x), 100).$$

Pour $r = 2$ et $n = 10000$, nous avons :

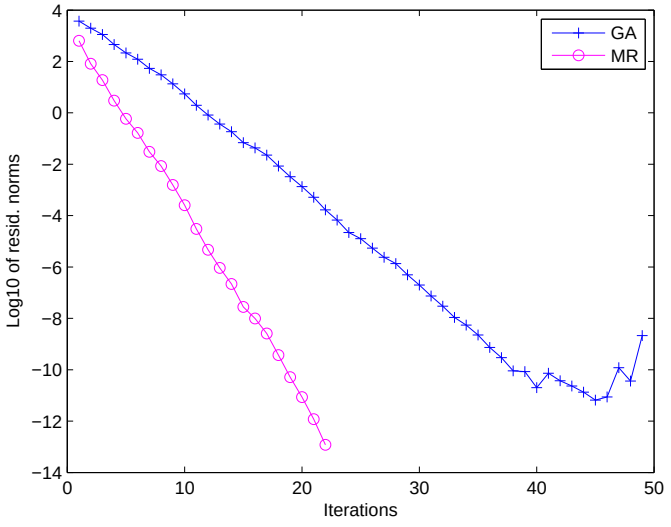


FIGURE 2.3: Résultats de l'exemple 2 : MR(PCGG)

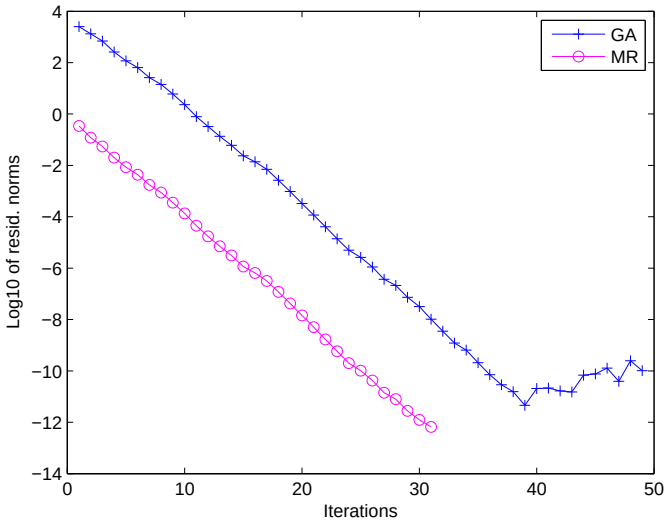


FIGURE 2.4: Résultats de l'exemple 2 : MR(GL-LSQR)

## 2.7 Conclusion

Dans ce chapitre, nous avons proposé deux méthodes itératives pour résoudre les équations matricielles de Lyapunov de grande taille. Ces méthodes sont basées sur la projection sur des sous espaces de Krylov étendus global. La première est basée sur la condition d'orthogonalité de Galerkin et la deuxième est basée sur la condition de minimisation de résidu MR. Le problème de minimisation réduit de la deuxième méthode MR est résolu par : la méthode LSQR global ou bien la méthode PCGG. Nous avons donné des exemples numériques pour montrer l'efficacité des deux méthodes proposées.

Chapitre 3

# L'Extended-Bloc Arnoldi algorithme avec minimisation de résidu pour les équations de Sylvester creuse et de grandes taille

Dans ce chapitre, nous présentons une nouvelle méthode pour résoudre des équations matricielles de Sylvester de grande taille. La méthode proposée est une méthode itérative basée sur la projection dans des sous-espaces de Krylov étendu (extended) par blocs et la minimisation de résidu (MR minimal residual). nous résolvons le problème réduit soit par des méthodes directes (méthode de Hu et Reichel [57]) ou bien des méthodes itératives (globale LSQR, PGCG).

## 3.1 Introduction

Dans ce chapitre, nous considérons l'équation matricielle de Sylvester

$$AX + XB + EF^T = 0 \tag{3.1}$$

où $A$ et $B$ sont des matrices de taille $n \times n$ et $s \times s$ respectivement, $E \in \mathbb{R}^{m\times n}$ et $F \in \mathbb{R}^{m\times n}$ sont des matrices de rang maximale, $X \in \mathbb{R}^{n\times s}$ est la matrice inconnue à déterminer.

Les équations matricielles de Sylvester jouent un rôle important dans de nombreuses théories : automatique, traitement du signal, restauration d'images, filtrage, réduction de modèle en théorie du contrôle, calcul d'un contrôle optimal en contrôle linéaire quadratique, discrétisation d'équations aux dérivées partielles, voir par exemple les références

suivantes [21, 39, 56, 62].

Pour résoudre l'équation matricielle de Sylvester, on distinguera deux cas :
Dans le cas où les matrices $A$ et $B$ sont de petite ou moyenne taille, il existe plusieurs méthodes directes dont par exemple la méthode Bartels-Stewart [11] et celle de Hessenberg-Schur [45]. Ces méthodes sont basées sur la réduction de Hessenberg de la plus grande des deux matrices $A$ et $B$ et la décomposition de Schur de la plus petite.

Les dernières années, plusieurs méthodes de projection basées sur les sous-espaces de Krylov ont été proposées, citons par exemple les références suivantes [36, 39, 56, 60, 62, 68, 82, 98, 102]. L'idée principale de ces méthodes est d'utiliser des sous-espaces de Krylov [57, 98] ou Krylov étendu par blocs ou global [55, 56, 64] ou sous-espace de Krylov rationnel [82] puis appliquer un algorithme de type Arnoldi pour construire une base orthonormale de l'espace de projection. L'équation matricielle de Sylvester de grande taille est alors projetée sur ces sous-espaces de Krylov. La solution de l'équation matricielle peut s'exprimer en fonction de la solution de l'équation projetée et des matrices orthogonales construites à partir les sous-espaces de Krylov. On peut citer aussi la méthode ADI (Alternating Directional Implicit) [21, 23, 78]. La méthode ADI (ou Lr-ADI) permet d'obtenir une convergence plus rapide sous certaines conditions de $A$ et $B$.

L'équation matricielle de Sylvester (3.1) peut s'écrire comme un système linéaire :

$$(I_s \otimes A + B^T \otimes I_n)vec(X) = -vec(EF^T), \tag{3.2}$$

où $I_p$ désigne la matrice identité de taille $p \times p$.

Cette écriture qui ramène l'équation de Sylvester à la résolution d'un système linéaire de taille $ns \times ns$, ne présente en réalité que peu d'avantage en terme de résolution, qu'elle soit directe ou itérative. Cette formulation permet d'établir le résultat suivant :

**Proposition 3.1.1.** *L'équation $AX + XB + EF^T = 0$ admet une solution unique si et seulement si les matrices $A$ et $-B$ n'ont aucune valeur propre commune, c'est-à-dire :*

$$\sigma(A) \cap \sigma(-B) = \emptyset.$$

*où $\sigma(A)$ et $\sigma(-B)$ représentent respectivement le spectre de $A$ et $-B$*

*Démonstration.* En effet, le système linéaire (3.2) possède une unique solution si et seulement si la matrice $I_n \otimes A + B^T \otimes I_m$ est inversible. Comme les valeurs propres de $I_n \otimes A + B^T \otimes I_m$ sont $(\lambda_i - \mu_j)$ où $\lambda_i$ et $\mu_j$ sont les valeurs propres de $A$ et $B^T$ respectivement. La valeur 0 ne sera pas dans le spectre de $I_n \otimes A + B^T \otimes I_m$ si et seulement si

$$\sigma(A) \cap \sigma(-B) = \emptyset.$$

□

Cette proposition donne une condition nécessaire et suffisante d'existence et d'unicité de la solution de l'équation de Sylvester (3.1). Des hypothèses plus fortes nous permettent de donner une écriture explicite de la solution de l'équation de Sylvester.

**Définition 3.1.2.** *Une matrice carrée est dite stable si toutes ses valeurs propres appartiennent au sous ensemble* $\mathbb{R}^*_- + i\mathbb{R}$ *de* $\mathbb{C}$.

**Proposition 3.1.3.** *Si les matrices* $A$ *et* $B$ *sont stables, alors l'équation (3.2) possède une unique solution* $X$*, de plus*

$$X = \int_0^{+\infty} e^{tA} E F^T e^{tB} dt.$$

La suite de ce chapitre est organisé de la façon suivante : dans un premier temps, nous allons rappeler la méthode de projection de Petrov-Galerkin sur le sous-espace de Krylov étendu (extended). Le paragraphe suivant est consacré à notre méthode : la méthode de la minimisation de résidu MR. Puis, nous exposons des méthodes itératives ou directes pour résoudre le problème de minimisation réduit. La Section 5 est consacrée à la forme factorisée de l'approximation de la solution approchée (low rank approximate solutions). Dans les Sections 6 et 7, nous donnons les algorithmes pour résoudre l'équation matricielle de Sylvester par l'approche de Galerkin et la méthode de la minimisation de résidu MR. Dans l'avant dernier paragraphe, nous appliquons notre approche MR, et la méthode de Galerkin à l'équation matricielle de Stein non Symétrique.

Finalement, dans le dernier paragraphe, nous donnons quelques résultats numériques qui nous permettront de comparer les méthodes de minimisation de résidu MR et la méthode de Galerkin ou la méthode Lr-ADI.

## 3.2 Méthode de type Galerkin

Dans cette section nous allons rappeler l'approche de Galerkin pour résoudre l'équation de Sylvester de grande taille (Pour plus de détails voir [56]). Cette approche est basée sur les sous-espaces de Krylov étendus (extended) par blocs et la condition de Galerkin.

Considérons l'équation matricielle de Sylvester

$$AX + XB + EF^T = 0, \tag{3.3}$$

où $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{s\times s}$ et $E,\ F \in \mathbb{R}^{n\times r}$ avec $r << n$. Les matrices $E$ et $F$ étant supposées avoir un rang $r$ petit par rapport aux tailles de $A$ et $B$. Nous allons chercher

à construire des approximations $X_m^{GA}$ de la solution $X$ sous la forme

$$X_m^{GA} = \mathbb{V}_m Y_m^{GA} \mathbb{W}_m^T, \tag{3.4}$$

où $\mathbb{V}_m$, $\mathbb{W}_m$ sont les matrices orthonormales construites en appliquant simultanément $m$ itérations de l'algorithme d'Arnoldi étendu par blocs aux paires de matrices $(A, E)$ et $(B^T, F)$ respectivement. Dans ce cas, on a

$$\mathbb{T}_A = \mathbb{V}_m^T A \mathbb{V}_m, \ \mathbb{T}_B = \mathbb{W}_m^T B \mathbb{W}_m.$$

La matrice $Y_m^{GA}$ est déterminée en imposant la condition d'orthogonalité

$$\mathcal{R}_m := \mathcal{A} X_m^{GA} + X_m^{GA} B - E F^T \perp \mathbb{L}_m(A, B),$$

où $\mathbb{L}_m(A, B)$ est le sous espace de $\mathbb{R}^{2mr \times 2mr}$ constitué des matrices de la forme $\mathbb{V}_m Y \mathbb{W}_m^T$. Donc, nous avons

$$\mathbb{V}_m^T \mathcal{R}_m \mathbb{W}_m = 0.$$

On montre facilement que $Y_m^{GA} \in \mathbb{R}^{mr \times mr}$ est la solution de l'équation de Sylvester de taille réduite

$$\mathbb{T}_m^A Y_m^{GA} + Y_m^{GA} {\mathbb{T}_m^B}^T + \widetilde{E}\widetilde{F}^T = 0, \tag{3.5}$$

où $\widetilde{E} = \mathbb{V}_m^T E$ et $\widetilde{F} = \mathbb{W}_m^T F$. La solution $Y_m^{GA}$ de l'équation de Sylvester de taille réduite (3.5) peut être obtenue par une méthode directe, par exemple la méthode Bartels-Stewart [11]. En pratique l'équation de Sylvester projetée (3.5) est résolue par la fonction " lyap" de Matlab. Pour tester la convergence, il est nécessaire de calculer la norme du résidu $\mathcal{R}_m$, pour cela le résultat suivant donne une expression du résidu $\mathcal{R}_m$ en fonction des matrices de petite taille.

**Théorème 3.2.1.** *[56] Soit $Y_m^{GA}$ la solution exacte de l'équation projetée (3.5). Alors la norme de Frobenius du résidu $\mathcal{R}_m$ est donnée par*

$$\| \mathcal{R}_m \|_F = \sqrt{\| Y_m \mathbb{E}_m (T_{m+1,m}^A)^T \|_F^2 + \| T_{m+1,m}^B \mathbb{E}_m^T Y_m \|_F^2} \tag{3.6}$$

L'intérêt pratique de ce résultat sur le résidu permet de tester la convergence dans l'algorithme sans avoir à calculer $X_m^{GA} = \mathbb{V}_m Y_m^{GA} \mathbb{W}_m^T$ à chaque itération.

## 3.3 Méthode de minimisation du résidu

Dans les dernières années, plusieurs méthodes de projection sur les sous-espaces de Krylov ont été proposées pour trouver les approximations $X_m$ de la solution exacte $X$ de l'équation Lyapunov, Riccati ou Sylvester ; voir par exemple, [39, 60, 62, 68, 98, 102]. La plupart de ces méthodes de projection utilisent la condition de Galerkin pour trouver la

solution approchée $X_m$.

Dans cette section, nous nous intéressons à la résolution de l'équation de Sylvester par la méthode de projection sur les sous-espaces de Krylov étendus avec la condition de minimisation du résidu MR (minimal residual). Donc nous allons chercher à construire des approximations $X_m^{MR}$ de la solution $X$ sous la forme

$$X_m^{MR} = \mathbb{V}_m Y_m^{MR} \mathbb{W}_m^T, \tag{3.7}$$

de telle sorte que $X_m^{MR}$ minimise le résidu $\mathcal{R}_m(X_m)$, où $X_m = \mathbb{V}_m Y_m \mathbb{W}_m^T$. Pour cela, nous nous intéressons à la résolution du problème de minimisation suivant :

$$X_m^{MR} = \arg \min_{X = \mathbb{V}_m Y_m \mathbb{W}_m^T} \left\| AX + XB + EF^T \right\|_F, \tag{3.8}$$

où $\mathbb{V}_m = [V_1, V_2, \cdots, V_m]$, $\mathbb{W}_m = [W_1, W_2, \cdots, W_m] \in \mathbb{R}^{n \times mr}$ sont les matrices orthonormales construites en appliquant simultanément $m$ itérations de l'algorithme d'Arnoldi étendu par blocs aux paires de matrices $(A, E)$ et $(B^T, F)$ respectivement. Dans ce cas, nous avons :

$$A\mathbb{V}_m = \mathbb{V}_{m+1} \bar{\mathbb{T}}_m^A \tag{3.9}$$

et

$$B^T \mathbb{W}_m = \mathbb{W}_{m+1} \bar{\mathbb{T}}_m^B. \tag{3.10}$$

Nous avons alors le résultat suivant

**Théorème 3.3.1.** *Soient $\mathbb{V}_m$ et $\mathbb{W}_m$ sont les matrices orthonormales construites en appliquant simultanément $m$ itérations de l'algorithme d'Arnoldi étendu par blocs aux paires de matrices $(A, E)$ et $(B^T, F)$ respectivement. Alors résoudre le problème de minimisation*

$$X_m^{MR} = \arg \min_{X_m = \mathbb{V}_m Y_m \mathbb{W}_m^T} \left\| AX_m + X_m B + EF^T \right\|_F$$

*revient à résoudre le problème suivant*

$$Y_m^{MR} = \arg \min_{Y_m} \left\| \bar{\mathbb{T}}_m^A Y_m \begin{pmatrix} I & 0 \end{pmatrix} + \begin{pmatrix} I \\ 0 \end{pmatrix} Y_m \bar{\mathbb{T}}_m^{BT} + \begin{pmatrix} R_E R_F^T & 0 \\ 0 & 0 \end{pmatrix} \right\|_F \tag{3.11}$$

*où $E = V_1 R_E$ et $F = W_1 R_F$ sont les décomposition $QR$ de $E$ et $F$ respectivement.*

*Démonstration.* Soient $E = V_1 R_E$ et $F = W_1 R_F$ sont les décomposition $QR$ de $E$ et $F$ respectivement. D'après les relations (3.9) et (3.10) nous avons

$$\begin{aligned}
&\min_{X=\mathbb{V}_m Y_m \mathbb{W}_m^T} \left\| AX + XB + EF^T \right\|_F \\
&= \min_{Y_m} \left\| A\mathbb{V}_m Y_m \mathbb{W}_m^T + \mathbb{V}_m Y_m \mathbb{W}_m^T B + V_1 R_E R_F^T W_1^T \right\|_F \\
&= \min_{Y_m} \left\| \mathbb{V}_{m+1} \bar{\mathbb{T}}_m^A Y_m \mathbb{W}_m^T + \mathbb{V}_m Y_m \bar{\mathbb{T}}_m^{BT} \mathbb{W}_{m+1}^T + V_1 R_E R_F^T W_1^T \right\|_F \\
&= \min_{Y_m} \left\| \mathbb{V}_{m+1} \left[ \bar{\mathbb{T}}_m^A Y_m \begin{pmatrix} I & 0 \end{pmatrix} + \begin{pmatrix} I \\ 0 \end{pmatrix} Y_m \bar{\mathbb{T}}_m^{BT} + \begin{pmatrix} R_E R_F^T & 0 \\ 0 & 0 \end{pmatrix} \right] \mathbb{W}_{m+1}^T \right\|_F \\
&= \min_{Y_m} \left\| \left[ \bar{\mathbb{T}}_m^A Y_m \begin{pmatrix} I & 0 \end{pmatrix} + \begin{pmatrix} I \\ 0 \end{pmatrix} Y_m \bar{\mathbb{T}}_m^{BT} + \begin{pmatrix} R_E R_F^T & 0 \\ 0 & 0 \end{pmatrix} \right] \right\|_F .
\end{aligned}$$

□

Notons que le calcul de la norme du résidu $\mathcal{R}(X_m^{MR})$ à chaque itération $m$ nécessite seulement la connaissance de $Y_m^{MR}$ sans avoir à construire la solution approximative $X_m^{MR}$. Cette norme du résidu $\mathcal{R}(X_m^{MR})$ peut être calculée par

$$r_m := \left\| \left[ \bar{\mathbb{T}}_m^A Y_m^{MR} \begin{pmatrix} I & 0 \end{pmatrix} + \begin{pmatrix} I \\ 0 \end{pmatrix} Y_m^{MR} \bar{\mathbb{T}}_m^{BT} + \begin{pmatrix} R_E R_F^T & 0 \\ 0 & 0 \end{pmatrix} \right] \right\|_F . \tag{3.12}$$

La solution approchée $X_m^{MR}$ est calculée que si la convergence est réalisée. Dans la section suivante, nous proposons des méthodes (directes ou itératives) pour calculer la solution $Y_m^{MR}$ du problème de minimisation réduit (3.11).

## 3.4 Méthodes pour résoudre le problème de minimisation réduit

### 3.4.1 La méthode directe basée sur la formulation de Kronecker

Dans ce paragraphe, nous nous intéressons à la résolution du problème (3.11), en utilisant les propriétés du produit de Kronecker, le problème (3.11) peut s'écrire comme un problème aux moindres carrés vectorielle :

$$\min_y \left\| \left[ \begin{pmatrix} I \\ 0 \end{pmatrix} \otimes \bar{\mathbb{T}}_m^A + \bar{\mathbb{T}}_m^B \otimes \begin{pmatrix} I \\ 0 \end{pmatrix} \right] y + c \right\|_F , \tag{3.13}$$

où $c = vec \begin{pmatrix} R_E R_F^T & 0 \\ 0 & 0 \end{pmatrix}$ et $y = vec(Y)$. Ce problème aux moindres-carrés est alors résolu par des méthodes directes ou itératives. Par exemple :

posons

$$H = \left[ \left( \begin{array}{c} I \\ 0 \end{array} \right) \otimes \bar{\mathbb{T}}_m^A + \bar{\mathbb{T}}_m^B \otimes \left( \begin{array}{c} I \\ 0 \end{array} \right) \right].$$

Si $H$ est de rang maximal, nous avons $y = -(H^T H)^{-1} H^T c = -H^\dagger c$ est la solution aux moindres carrés, où $H^\dagger$ désigne le pseudo-inverse de la matrice $H$, et le résidu est donner par $r_m = Hy + c = (I - H(H^T H)^{-1} H^T)c$.

Pour un petit nombre d'itérations $m$, cette méthode donne une bonne approximation. Toutefois, lorsque $m$ augmente, la méthode est très lente. Dans les prochaines sous-sections, nous exposons d'autre méthodes efficaces pour résoudre le problème de minimisation (3.11).

### 3.4.2 Méthode de Hu et Reichel

Nous exposons dans cette partie la méthode de Hu et Reichel (voir [57]) pour résoudre le problème réduit (3.11). Plus précisément, nous appliquons l'approche donnée dans [74] et la stratégie donnée dans [102].

Écrivons les matrices $\overline{\mathbb{T}}_A$ et $\overline{\mathbb{T}}_B$ sous la forme :

$$\overline{\mathbb{T}}_A = \left( \begin{array}{c} \mathbb{T}_A \\ h_A \end{array} \right) \text{ et } \overline{\mathbb{T}}_B = \left( \begin{array}{c} \mathbb{T}_B \\ h_B \end{array} \right)$$

où $h_A$ et $h_B$ représentent les $2r$ dernières lignes des matrices $\overline{\mathbb{T}}_A$ et $\overline{\mathbb{T}}_B$ respectivement. Soient $\mathbb{T}_A = U T_A U^T$ et $\mathbb{T}_B = V T_B V^T$ la décomposition de Schur de $\mathbb{T}_A$ et $\mathbb{T}_B$ respectivement.

Alors le problème de minimisation est équivalent à

$$\begin{aligned}
&\min_{Y_m \in \mathbb{R}^{2mr \times 2mr}} \left\| \left[ \bar{\mathbb{T}}_m^A Y_m \left( \begin{array}{cc} I & 0 \end{array} \right) + \left( \begin{array}{c} I \\ 0 \end{array} \right) Y_m \bar{\mathbb{T}}_m^{BT} + \left( \begin{array}{cc} R_E R_F^T & 0 \\ 0 & 0 \end{array} \right) \right] \right\|_F \\
&= \min_{Y_m \in \mathbb{R}^{2mr \times 2mr}} \left\| \left[ \left[ \begin{array}{c} T_A \\ h_A \end{array} \right] Y_m \left( \begin{array}{cc} I & 0 \end{array} \right) + \left( \begin{array}{c} I \\ 0 \end{array} \right) Y_m \left[ \begin{array}{c} T_B \\ h_B \end{array} \right]^T + \left( \begin{array}{cc} R_E R_F^T & 0 \\ 0 & 0 \end{array} \right) \right] \right\|_F, \\
&= \min_{Y_m \in \mathbb{R}^{2mr \times 2mr}} \left\| \left( \begin{array}{cc} T_A Y + Y T_B^T + R_E R_F^T & 0 \\ h_A Y & Y h_B^T \end{array} \right) \right\|_F
\end{aligned}$$

ce qui est équivalent à

$$\begin{aligned}
&\min_{Y_m \in \mathbb{R}^{2mr \times 2mr}} \left( \left\| \mathbb{T}_A Y + Y \mathbb{T}_B^T + R_E R_F^T \right\|_F^2 + \| h_A Y \|_F^2 + \left\| Y h_B^T \right\|_F^2 \right) \\
&= \min_{Y_m \in \mathbb{R}^{2mr \times 2mr}} \left( \left\| U T_A U^T Y + Y (V T_B V^T)^T + R_E R_F \right\|_F^2 + \| h_A Y \|_F^2 + \left\| Y h_B^T \right\|_F^2 \right) \\
&= \min_{\widetilde{Y}_m \in \mathbb{R}^{2mr \times 2mr}} \left( \left\| T_A \widetilde{Y} + \widetilde{Y} T_B^T + U^T R_E R_F^T V \right\|_F^2 + \left\| h_A U \widetilde{Y} \right\|_F^2 + \left\| \widetilde{Y} V^T h_B^T \right\|_F^2 \right), \quad \text{où} \quad \widetilde{Y} = U^T Y V
\end{aligned}$$

On utilisant le produit de Kronecker nous obtenons

$$\min_{Y\in\mathbb{R}^{2mr\times 2mr}} \left\| \begin{bmatrix} I\otimes T_A + T_B\otimes I \\ I\otimes h_A U \\ h_B V\otimes I \end{bmatrix} Vect(\widetilde{Y}) + \begin{bmatrix} Vect(U^T R_E R_F^T V) \\ 0 \\ 0 \end{bmatrix} \right\|_2^2 \tag{3.14}$$

Nous pouvons écrire le problème (3.14) de la façon suivante

$$\min_{Y} \left\| \begin{bmatrix} R \\ S \end{bmatrix} y + \begin{bmatrix} d \\ 0 \end{bmatrix} \right\|_2^2 \tag{3.15}$$

où $R = I\otimes T_A + T_B\otimes I$, $S = \begin{bmatrix} I\otimes h_A U \\ h_B V\otimes I \end{bmatrix}$, $y = Vect(\widetilde{Y})$ et $d = Vect(U^T R_E R_F^T V)$.
L'équation normale associée est

$$(R^T R + S^T S)y + R^T d = 0. \tag{3.16}$$

Si la matrice $R$ est inversible, alors nous avons

$$(I + (SR^{-1})^T SR^{-1})\widetilde{y} + d = 0, \text{où} \quad y = R^{-1}\widetilde{y}.$$

Maintenant, en utilisant la formule de Sherman-Morrison-Woodbury, nous obtenons

$$\widetilde{y} = -d + (SR^{-1})^T (I + SR^{-1}(SR^{-1})^T)^{-1} SR^{-1} d.$$

Par conséquent, à partir de $\widetilde{y}$, nous obtenons facilement $Y_m$ la solution du problème (3.11).

Pour calculer $SR^{-1}$ on utilise la stratégie suivante :

$$SR^{-1} = \begin{pmatrix} I\otimes h_A U \\ h_B V\otimes I \end{pmatrix} (I\otimes T_A + T_B\otimes I)^{-1} = \begin{pmatrix} (I\otimes h_A U)(I\otimes T_A + T_B\otimes I)^{-1} \\ (h_B V\otimes I)(I\otimes T_A + T_B\otimes I)^{-1} \end{pmatrix}$$

Posons que

$$\begin{pmatrix} Z_1 \\ Z_2 \end{pmatrix} = \begin{pmatrix} (I\otimes h_A U)(I\otimes T_A + T_B\otimes I)^{-1} \\ (h_B V\otimes I)(I\otimes T_A + T_B\otimes I)^{-1} \end{pmatrix}$$

Alors

$$Z_1 = (I\otimes h_A U)(I\otimes T_A + T_B\otimes I)^{-1} \text{ et } Z_2 = (h_B V\otimes I)(I\otimes T_A + T_B\otimes I)^{-1}$$

Donc

$$(I\otimes T_A + T_B\otimes I)^T Z_1^T = (I\otimes (h_A U)^T).$$

Pour calculer $Z_1^T$, il suffit de résoudre le système suivant $(I \otimes T_A + T_B \otimes I)^T z = (I \otimes (h_A U)^T) e_i$ pour $i = 1, ..., 4mr^2$. Nous remarquons que $I \otimes (h_A U)^T$ est une matrice diagonale par blocs, avec tous les éléments diagonaux égaux à ceux de la matrice $(h_A U)^T$. Par conséquent, chaque colonne $h_l := (I \otimes (h_A U)^T) e_l$, $l = 1, ..., 4mr^2$, satisfait à la relation $Vec((h_A U)^T e_i e_j^T) = h_{i+2r(j-1)}$, $i = 1, ..., 2r$, $j = 1, ..., 2mr$.
Donc le système

$$(I \otimes T_A + T_B \otimes I)^T z = (I \otimes (h_A U)^T) e_l$$

est équivalent à l'équation de Sylvester

$$T_A Z + Z T_B = (h_A U)^T e_i e_j^T$$

avec $z = Vect(Z)$, pour $i = 1, ..., 2r$, $j = 1, ..., 2mr$, avec $l = i + 2r(j-1)$. Donc pour calculer $Z_1^T$ il suffit de résoudre les $2mr^2$ équations de Sylvester suivantes :

$$T_A Z + Z T_B = (h_A U)^T e_i e_j^T.$$

De la même manière, pour le deuxième bloc de $Z_2$. Donc pour calculer $Z_2^T$ il suffit de résoudre les $2mr^2$ équations de Sylvester suivantes :

$$T_A Z + Z T_B = (e_j^T h_B V) e_i.$$

La méthode de Hu et Reichel est résumé dans l'algorithme suivant :

**Algorithm 3** Algorithme de Hu-Reichel pour résoudre le problème (3.14)

1. Calculer la décomposition de Schur de $T_A$ et $T_B$, $T_A = U \mathbb{T}_A U^T$ et $T_B = U \mathbb{T}_B U^T$
2. Calculer $d = Vect(U^T R_E R_F^T V)$ :
3. Calculer $SR^{-1}$ par la résolution de $2mr$ équations de Sylvester
4. Résoudre $\widetilde{y} = -d + (SR^{-1})^T (I + SR^{-1}(SR^{-1})^T)^{-1} SR^{-1} d$.
5. Récupérer $Y_m^{MR}$

On peut également résoudre le problème réduit (3.14) par la méthode LSQR globale ou la méthode du gradient conjugué globale préconditionné.

**Remarque 3.4.1.** *La méthode aux moindres carrés par Kronecker et la méthode de Hu et Reichel nécessitent $O(m^3 r^3)$ multiplications qui est le même ordre que pour la méthode de Galerkin avec l'utilisation de l'algorithme Bartels-Stewart pour résoudre l'équation de Sylvester projetée (3.2). Lorsque m augmente, l'approche des moindres carrés et la Méthode Hu-Reichel deviennent très lentes.*

Dans les deux sections suivantes, nous exposons des méthodes itératives pour résoudre le problème de minimisation réduit.

### 3.4.3 La Méthode LSQR globale

A chaque étape du processus $m$, nous déterminons des approximations de la solution $Y_m^{MR}$ du problème de minimisation réduit (3.11).
Le problème de minimisation réduit (3.11) s'écrit de la manière suivante

$$\min_Y \|\mathcal{L}_m(Y) - \mathcal{C}\|_F, \tag{3.17}$$

où

$$\mathcal{L}_m(Y) = \bar{\mathbb{T}}_m^A Y \begin{pmatrix} I & 0 \end{pmatrix} + \begin{pmatrix} I \\ 0 \end{pmatrix} Y \bar{\mathbb{T}}_m^{BT} \tag{3.18}$$

et

$$\mathcal{C} = - \begin{pmatrix} R_E R_F^T & 0 \\ 0 & 0 \end{pmatrix}.$$

Nous appliquons la méthode la méthode LSQR Globale pour résoudre le problème de minimisation (3.17) avec l'opérateur adjoint de $\mathcal{L}_m$ par rapport au produit scalaire de Frobenius donné par

$$\mathcal{L}_m{}^T(Y) = (\bar{\mathbb{T}}_m^A)^T Y \begin{pmatrix} I \\ 0 \end{pmatrix} + \begin{pmatrix} I & 0 \end{pmatrix} Y \bar{\mathbb{T}}_m^B. \tag{3.19}$$

L'algorithme global LSQR permet de calculer une solution approchée $Y^k$ avec $1 \leq k \leq kmax$, du problème de minimisation (3.17 ). Afin de savoir si une solution $X_m^{MR}$ approche suffisamment la solution exacte $X$, nous devons être en mesure de calculer de façon simple et peu couteuse le résidu $\mathcal{R}_m$ associé à $X_m^{MR}$. Ensuite, nous donnons l'expression de $\|\mathcal{R}_m(X_m)\|$ selon la norme du résidu associé à l'algorithme LSQR globale (Gl-LSQR) appliquée au problème de minimisation réduit.

**Théorème 3.4.2.** *Soit $X_m^{MR} = \mathbb{V}_m Y_m^{MR} \mathbb{W}_m^T$ la solution approchée de l'équation matricielle de Sylvester obtenu après m itérations de l'algorithme d'Arnoldi étendu par blocs, où*

$$Y_m^{MR} = \arg\min_Y \left\| \bar{\mathbb{T}}_m^A Y \begin{bmatrix} I & 0 \end{bmatrix} + \begin{bmatrix} I \\ 0 \end{bmatrix} Y \bar{\mathbb{T}}_m^{BT} + \begin{bmatrix} R_E R_F^T & 0 \\ 0 & 0 \end{bmatrix} \right\|_F. \tag{3.20}$$

*Alors,*

$$\|\mathcal{R}_m(X_m^{MR})\|_F = \sqrt{m}\Phi$$

*où $\Phi = \mid \overline{\Phi}_{kmax} \mid$ est donné dans l'algorithme 3 (Gl-LSQR).*

*Démonstration.* nous avons

$$\begin{aligned}
\|\mathcal{R}_m(X_m^{MR})\|_F &= \|AX_m^{MR} + X_m^{MR}B + EF^T\|_F \ \text{ où } \ X_m^{MR} = \mathbb{V}_m Y_m^{MR}\mathbb{W}_m^T \\
&= \left\| \bar{\mathbb{T}}_m^A Y_m^{MR} \begin{bmatrix} I & 0 \end{bmatrix} + \begin{bmatrix} I \\ 0 \end{bmatrix} Y_m^{MR}\bar{\mathbb{T}}_m^{BT} + \begin{bmatrix} R_E R_F^T & 0 \\ 0 & 0 \end{bmatrix} \right\|_F \\
&= \|(\beta_1 e_1 - \bar{T}_k z_k) \otimes I_m\|_F \\
&= \sqrt{m}\|(\beta_1 e_1 - \bar{T}_k z_k)\|_2 \\
&= \sqrt{m}\Phi.
\end{aligned}$$

□

Le Théorème 3.4.2 joue un rôle très important dans la pratique ; il permet de calculer la norme du résidu sans calculer la solution approchée pour tester la convergence. Des hypothèses plus fortes nous permettent de donner une écriture de la solution de l'équation de Sylvester et une majoration en norme de l'erreur $X - X_m$.
Si les matrices $A$ et $B$ sont stables ($Re(\lambda_i(A)) < 0$ et $Re(\lambda_i(B)) < 0$), alors l' équation de Sylvester (1.1) a une solution unique donnée par la représentation intégrale suivante (voir [80])

$$X = \int_0^\infty e^{tA} E F^T e^{tB} dt.$$

La norme logarithmique "2-norme" de la matrice stable $A$ est définies par

$$\mu_2(A) = \frac{1}{2}\,\lambda_{max}(A + A^T) \ < 0.$$

La norme logarithmique donne une majoration de l'exponentielle de matrices. Il est connu que (voir [90]).

$$\| \, e^{tA} \, \|_2 \leq \, e^{\mu_2(A)t}; \ \ t \geq 0. \tag{3.21}$$

Dans le théorème suivant et avec les mêmes notations que dans le théorème précédent, nous donnons une majoration en norme de l'erreur $X - X_m$

**Théorème 3.4.3.** *Supposons que $A$ et $B$ soient des matrices stables et soit $X_m^{MR} = \mathbb{V}_m Y_m^{MR}\mathbb{W}_m^T$ la solution approchée de l'équation de Sylvester obtenu après $m$ itérations de l'algorithme d'Arnoldi étendu par blocs. Alors nous avons*

$$\| \, X - X_m^{MR} \, \|_2 \leq \frac{-r_m}{\mu_2(A) + \mu_2(B)}, \tag{3.22}$$

*où $r_m$ est la norme du résidu donné par (3.12).*

*Démonstration.* Rappelons que $\mathcal{R}(X_m^{MR})$ correspond au résidu

$$AX_m^{MR} + X_m^{MR}B + EF^T = R(X_m^{MR})$$

de l'équation matricielle de Sylvester :

$$AX + XB + EF^T = 0$$

La soustraction terme à terme permet de retrouver l'équation suivante :

$$A(X_m^{MR} - X) + (X_m^{MR} - X)B = \mathcal{R}(X_m^{MR}). \tag{3.23}$$

Comme $A$ et $B$ sont supposés être stable, nous trouvons

$$X_m^{MR} - X = \int_0^\infty e^{tA}\mathcal{R}(X_m^{MR})e^{tB}dt.$$

Par conséquent, en utilisant la norme logarithmique, on obtient

$$\| X - X_m^{MR} \| \leq \| \mathcal{R}(X_m^{MR}) \|_2 \int_0^\infty e^{t(\mu_2(A)+\mu_2(B))}\, dt.$$

Comme $\| \mathcal{R}(X_m^{MR}) \|_2 \leq \| \mathcal{R}(X_m^{MR}) \|_F$ et $(\mu_2(A) + \mu_2(B)) < 0$, nous obtenons :

$$\| X - X_m^{MR} \|_2 \leq \frac{-r_m}{\mu_2(A) + \mu_2(B)},$$

où $r_m$ ($r_m = \| \mathcal{R}(X_m^{MR}) \|_F$) est la norme du résidu donné par (3.12). □

La convergence de l'algorithme LSQR globale (Gl-LSQR) peut être lente. Ensuite nous devons utiliser cette méthode avec un préconditionneur. Pour cela, il existe plusieurs stratégies que nous pouvons utiliser, on pourrait adapter par exemple les techniques utilisées dans [7].

### 3.4.4 La méthode du gradient conjugué globale préconditionné

Dans cette section, nous considérons la méthode du gradient conjugué globale préconditionnés (PGCG) pour résoudre le problème de minimisation réduit (3.11). La méthode du gradient conjugué préconditionné (PCG) classique appliquée à l'équation normale. L'équation normale associé à (3.11) est donnée par

$$\mathcal{L}_m^*(\mathcal{L}_m Y)) = \mathcal{L}_m^*(\mathcal{C}), \tag{3.24}$$

où $\mathcal{L}_m$ et $\mathcal{L}_m^*$ les opérateur donné par (3.18) et (3.19) respectivement.

Soient les matrices $\overline{\mathbb{T}}_m^A$ et $\overline{\mathbb{T}}_m^B$ sous la forme

$$\overline{\mathbb{T}}_m^A = \begin{pmatrix} \mathbb{T}_m^A \\ h_m^A \end{pmatrix} \text{ et } \overline{\mathbb{T}}_B = \begin{pmatrix} \mathbb{T}_m^B \\ h_m^B \end{pmatrix},$$

où $h_A$ et $h_B$ représentent les $2r$ dernières lignes de la matrice $\overline{\mathbb{T}}_m^A$ et $\overline{\mathbb{T}}_m^B$, respectivement. Par conséquent, l'équation normale (3.24) peut s'écrire comme suit

$$\overline{\mathbb{T}}_m^{A\,T}\overline{\mathbb{T}}_m^A Y + Y\overline{\mathbb{T}}_m^{B\,T}\overline{\mathbb{T}}_m^B + \mathbb{T}_m^{A\,T} Y \mathbb{T}_m^{B\,T} + \mathbb{T}_m^A Y \mathbb{T}_m^B - \mathcal{C}_1 = 0, \tag{3.25}$$

où $\mathcal{C}_1 = \mathcal{L}_m^T(\mathcal{C})$. Considérons la décomposition en valeurs singulières (SVD) de $\overline{\mathbb{T}}_m^A$ et $\overline{\mathbb{T}}_m^B$ :

$$\overline{\mathbb{T}}_m^A = \overline{U}_A \overline{\Sigma}_A \overline{V}_A^T; \qquad \overline{\mathbb{T}}_m^B = \overline{U}_B \overline{\Sigma}_B \overline{V}_B^T, \tag{3.26}$$

alors nous obtenons la décomposition suivante

$$\overline{\mathbb{T}}_m^{A\,T}\overline{\mathbb{T}}_m^A = Q_A D_A Q_A^T, \qquad \overline{\mathbb{T}}_m^{B\,T}\overline{\mathbb{T}}_m^B = Q_B D_B Q_B^T, \tag{3.27}$$

où $Q_A = \overline{V}_A$, $Q_B = \overline{V}_B$ et $D_A = \overline{\Sigma}_A^T \overline{\Sigma}_A$.

Soient $\widetilde{Y} = Q_A^T Y Q_B$ et $\widetilde{\mathcal{C}} = Q_A^T \mathcal{C}_1 Q_B$, donc l'équation normale (3.25) peut aussi s'écrire

$$D_A \widetilde{Y} + \widetilde{Y} D_B + \widetilde{\mathbb{T}}_m^A \widetilde{Y} \widetilde{\mathbb{T}}_m^B + (\widetilde{\mathbb{T}}_m^A)^T \widetilde{Y} \widetilde{\mathbb{T}}_m^B)^T - \widetilde{\mathcal{C}} = 0, \tag{3.28}$$

où $\widetilde{\mathbb{T}}_m^A = Q_A^T \mathbb{T}_m^A Q_A$, $\widetilde{\mathbb{T}}_m^B = Q_B^T \mathbb{T}_m^B Q_B$ et $\widetilde{Y} = Q_A^T Y Q_B$. Cette expression propose que l'on peut utiliser l'opérateur matriciel suivant comme un préconditionneur

$$\mathcal{P}(\widetilde{Y}) = D_A \widetilde{Y} + \widetilde{Y} D_B. \tag{3.29}$$

On peut voir que l'expression (3.28) correspondant à l'équation normale de l'opérateur matriciel suivant

$$\widetilde{\mathcal{L}}_m(\widetilde{Y}) = \widetilde{\mathcal{T}}_m^A \widetilde{Y} \begin{pmatrix} Q_B^T & 0 \end{pmatrix} + \begin{pmatrix} Q_A \\ 0 \end{pmatrix} \widetilde{Y} (\widetilde{\mathcal{T}}_m^B)^T \tag{3.30}$$

où $\widetilde{\mathcal{T}}_m^A = \overline{\mathbb{T}}_m^A Q_A$ et $\widetilde{\mathcal{T}}_m^B = \overline{\mathbb{T}}_m^B Q_B$. Par conséquent, l'algorithme du gradient conjugué préconditionné globale est obtenu en appliquant le préconditionneur (3.29) à l'équation normale associée à l'opérateur matriciel défini par (3.30). Ceci est résumé dans l'algorithme suivant :

**Algorithme 3 : Algorithme du gradient conjugué globale préconditionné(PGCG)**

1. On choisit une tolérance $tol > 0$, et un nombre maximal d'itérations $j_{max}$ ;
   Choisir $\widetilde{Y}_0$ et on fixe $\widetilde{R}_0 = \mathcal{C} - \widetilde{\mathcal{L}}_m(\widetilde{Y}_0)$ ; $S_0 = \widetilde{\mathcal{L}}_m^*(\widetilde{R}_0)$, $Z_0 = \mathcal{P}^{-1}(S_0)$ ; $P_0 = S_0$.
2. **Pour** $j = 0, 1, 2, ..., j_{max}$
   (a) $W_j = \widetilde{\mathcal{L}}_m(P_j)$
   (b) $\alpha_j = \langle S_j, Z_j \rangle_F / |W_j|_F^2$
   (c) $\widetilde{Y}_{j+1} = \widetilde{Y}_j + \alpha_j P_j$
   (d) $\widetilde{R}_{j+1} = \widetilde{R}_j - \alpha_j W_j$
   (e) **Si** $\|\widetilde{R}_{j+1}\|_F < tol$. **Arrêt**
   **Sinon**
   (f) $S_{j+1} = \widetilde{\mathcal{L}}_m^*(\widetilde{R}_{j+1})$
   (g) $Z_{j+1} = \mathcal{P}^{-1}(S_{j+1})$
   (h) $\beta_j = \langle S_{j+1}, Z_{j+1} \rangle_F / \langle S_j, Z_j \rangle_F$
   (i) $P_{j+1} = Z_{j+1} + \beta_j P_j$.
3. **Fin Pour**.

On note que l'utilisation du préconditionneur $\mathcal{P}$ nécessite à chaque itération la résolution d'une équation de Sylvester. Comme les matrices $D_A$ et $D_B$ de ces équations matricielles de Sylvester sont des matrices diagonales, ceci réduit les coûts de calcul.

## 3.5 Forme factorisée de l'approximation de la solution

On pourra décomposer la solution $X_m$ comme produit de deux matrices de rang inférieur, c'est-à-dire $X_m = Z_1 Z_2^T$ où $Z_1$ et $Z_2$ sont des matrices de petite rang (inférieure à $2mr$). Pour cela, nous considérons la décomposition en valeurs singulières de la matrice $Y_m \in \mathbb{R}^{2mr \times 2mr}$.

$$Y_m = U_1 \Sigma U_2^T$$

où $\Sigma$ est la matrice diagonale des valeurs singulières de $Y_m$ rangées dans l'ordre décroissant, $U_1$ et $U_2$ étant deux matrices unitaires. Nous fixons une certaine tolérance *dtol* et définissons $U_{1,l}$ et $U_{2,l}$ les matrices constituées des $l$ premières colonnes de $U_1$ et de $U_2$ correspondants aux $l$ valeurs singulières supérieures ou égales à *dtol*. Nous obtenons la décomposition en valeur singulière tronquée

$$Y_m \approx U_{1,l} \, \Sigma_l \, {U_{2,l}}^T$$

où $\Sigma_l = \text{diag}[\sigma_1, \ldots, \sigma_l]$. Soit $Z_{1,m} = \mathbb{V}_m U_{1,l} \Sigma_l^{1/2}$, et $Z_{2,m} = \mathbb{W}_m U_{2,l} \Sigma_l^{1/2}$, nous obtenons alors l'approximation de $X_m$ sous la forme d'une expression factorisée

$$X_m \approx Z_{1,m} Z_{2,m}^T. \tag{3.31}$$

Cette factorisation est très importante pour les problèmes de grande dimension, quand on n'a pas besoin de calculer l'approximation de la solution $X_m$, mais on a besoin de la stocker à chaque itération.

## 3.6 L'algorithme GA-LRSE

Dans cette section, nous présentons l'algorithme pour résoudre l'équation de Sylvester de grande taille **(GA-LRSE)** par la méthode de projection de Galerkin sur les sous-espaces de Krylov étendus (extended) par blocs.

**Algorithme 3 : pour résoudre l'équation de Sylvester (GA-Lr-SE)**

1. On choisit une tolérance $\epsilon > 0$ et un nombre maximal d'itérations $m_{max}$
2. **Pour** $m = 1, 2, 3, ..., m_{max}$
3. Construire les matrices $\mathbb{V}_m$ et $\mathbb{T}_m^A$ par l'algorithme 1 appliqué au couple $(A, E)$.
4. Construire les matrices $\mathbb{W}_m$ et $\mathbb{T}_m^B$ par l'algorithme 1 appliqué au couple $(B^T, F)$.
5. Résoudre l'équation de Sylvester projetée suivante

$$\mathbb{T}_m^A Y_m^{GA} + Y_m^{GA} (\mathbb{T}_m^B)^T + \widetilde{E}\widetilde{F}^T = 0$$

6. Calculer la norme de Frobenius du résidu :

$$\| \mathcal{R}_m \|_F = \sqrt{\| Y_m^{GA} \mathbb{E}_m (T_{m+1,m}^B)^T \|_F^2 + \| T_{m+1,m}^A \mathbb{E}_m^T Y_m^{GA} \|_F^2}$$

   - **Si** $\|\mathcal{R}_m\| < \epsilon$ , **Aller à 8**
   - **Sinon** $m = m + 1$
   - **Fin si**

7. **Fin pour** $m$
8. Calculer la décomposition en valeurs singulières (SVD) de la matrice $Y_m^{GA}$, c'est à dire $Y_m^{GA} = \mathcal{V}\Sigma\mathcal{W}^T$, où $\Sigma = diag(\sigma_1, \sigma_2, ..., \sigma_{2mr})$ et $\sigma_1 \geq \sigma_2 \geq ... \geq \sigma_{2mr}$ ;
   trouver $l$ telle que $\sigma_{l+1} \leq tol_{trn} < \sigma_l$ et soit $\Sigma_l = diag(\sigma_1, \sigma_2, ..., \sigma_l)$ ;
   $Z_m^A = \mathbb{V}_m \mathcal{V}_l \Sigma_l^{1/2}$, et$Z_m^B = \mathbb{W}_m \mathcal{W}_l \Sigma_l^{1/2}$
9. La solution approchée $X_m$ est donnée par $X_m \approx Z_m^A Z_m^{B^T}$.

## 3.7 L'algorithme MR-LR-Sylvester

Dans cette section, nous présentons l'algorithme pour résoudre l'équation de Sylvester de grande taille **(MR-Lr-SE)** par la méthode de minimisation du résidu sur les sous-espaces de Krylov étendus (extended) par blocs.

**Algorithme 4 : pour résoudre l'équation de Sylvester (MR-Lr-SE)**

1. On choisit une tolérance $tol > 0$ et un nombre maximum d'itérations $iter_{max}$.
2. Pour $m = 1, 2, 3, ..., iter_{max}$
3. Construire les matrices $\mathbb{V}_m$ et $\overline{\mathbb{T}}_m^A$ par l'algorithme (EBA) appliqué au couple $(A,\ E)$.
4. Construire les matrices $\mathbb{W}_m$ et $\overline{\mathbb{T}}_m^B$ par l'algorithme (EBA) appliqué au couple $(B^T,\ F)$.
5. Résoudre le problème de minimisation

$$Y_m^{MR} = \arg\min_{Y_m} \left\| \bar{\mathbb{T}}_m^A Y_m \begin{bmatrix} I & 0 \end{bmatrix} + \begin{bmatrix} I \\ 0 \end{bmatrix} Y_m \left(\bar{\mathbb{T}}_m^B\right)^T + \begin{bmatrix} R_E R_F^T & 0 \\ 0 & 0 \end{bmatrix} \right\|_F .$$

6. Si $\|\mathcal{R}_m\|_F \leq tol$, Arrêt,
7. La solution $X_m$ est donnée par $X_m \approx Z_{1,m}\, Z_{2,m}^T$.

## 3.8 Résolution de l'équation de Stein non symétrique

Dans ce paragraphe, nous considérons l'équation de Stein non symétrique

$$AXB - X + EF^T = 0, \tag{3.32}$$

où $A$ et $B$ sont des matrices carrées de taille $n \times n$ et $s \times s$, respectivement. Les matrices $E$ et $F$ sont de dimension $n \times r$ et $r \times s$ respectivement. Les matrices $A$ et $B$ sont supposées creuses de très grande taille $r << n,\ s$.

L'équation matricielle (3.32) (également appelée équation généralisée de Stein ou équation de Sylvester discrète) joue un rôle important dans la théorie du contrôle, la restauration d'images, filtrage de systèmes dynamiques non linéaire à temps discret de grande taille, à chaque étape de la méthode de Newton pour résoudre les équations algébriques de Riccati discrète et d'autres problèmes. Pour plus d'informations, voir les références suivantes [17, 18, 32, 33, 42, 66, 80].

Les méthodes directes pour résoudre l'équation matricielle de Stein non symétrique sont proposés dans [8, 11, 45, 98]. Ces méthodes sont intéressantes lorsque les matrices $A$ et $B$ sont de petite taille. L'équation matricielle (3.32) peut être formulée comme un système linéaire de taille $ns \times ns$ :

$$(B^T \otimes A - I_s \otimes I_n)vec(X) = -vec(EF^T), \tag{3.33}$$

On peut utiliser les méthodes de type Krylov pour résoudre le système linéaire (3.33), par exemple l'algorithme de GMRES [99]. Lorsque les matrices $A$ et $B$ sont de grande taille et creuses, cette approche ne peuvent être appliquées directement. Dans ce cas, il existe des méthodes pour résoudre l'équation matricielle de Stein non symétrique, par exemple la méthode low-rank Stein block-Arnoldi. Cette méthode basée sur la projection de l'équation (3.32) dans les sous-espaces de Krylov par blocs [63, 66].

L'équation matricielle de Stein non symétrique admet solution unique si et seulement si $\lambda_i(A)\lambda_j(B) \neq 1$ pour tout $i = 1, 2, ..., n$ et $j = 1, 2, ..., s$, où $\lambda_i(A)$ est la $i$-ème valeur propre de la matrice $A$. En particulier, si $\rho(A) < 1$ et $\rho(B) < 1$, où $\rho(A)$ désigne le rayon spectral de la matrice $A$, alors l'équation (3.33) admet une solution unique $X$. De plus, elle peut s'exprimer sous la forme d'une série matricielle suivante :

$$X = \sum_{i=0}^{\infty} A^i \, E \, F^T \, B^i.$$

Nous nous référons à [80] pour les démonstrations de ces résultats. Nous supposons que la conditions d'existence et d'unicité sont vérifiées pour chacune des équations de Stein rencontrées dans la suite de cette section. La méthode proposée par Jbilou (low-rank Stein block-Arnoldi) permet de construire des approximations $X_m$ de petit rang de la forme

$$X_m = \mathcal{V}_m Y_m \mathcal{W}_m^T$$

où $\mathcal{V}_m$, $\mathcal{W}_m$ sont les matrices orthonormales construites en appliquant simultanément $m$ itérations de l'algorithme d'Arnoldi par blocs aux paires de matrices $(A, E)$ et $(B^T, F)$ respectivement, et $Y_m$ la solution de l'équation de Stein projetée suivante :

$$\mathcal{H}_{\mathcal{A}} Y_m \mathcal{H}_{\mathcal{B}}^T - Y_m + \widetilde{\mathcal{E}}\widetilde{\mathcal{F}}^T = 0, \tag{3.34}$$

où $\widetilde{\mathcal{E}} = \mathcal{V}_m^T \mathcal{E}$ et $\widetilde{\mathcal{F}} = \mathcal{W}_m^T \mathcal{F}$.

Dans cette section, nous exposons ensuite : la même méthode de Jbilou mais dans les sous-espaces de Krylov étendus (extended) par blocs, et la méthode de la minimisation du résidu (MR) pour résoudre l'équation de Stein non symétrique.

### 3.8.1 Méthode de Galerkin

Dans cette section nous allons rappeler l'approche de Galerkin pour résoudre l'équation de Stein non symétrique de grande taille. Cette approche a été introduite par Jbilou dans [63]. On considère l'équation de Stein non symétrique (3.32). Nous supposons que les matrices $E$ et $F$ sont de rang maximal, nous allons chercher à construire une suite des approximations de petit rang $(X_m^{GA})_{m\in\mathbb{N}^*}$ de la solution $X$ sous la forme

$$X_m^{GA} = \mathbb{V}_m Y_m^{GA} \mathbb{W}_m^T, \tag{3.35}$$

où $\mathbb{V}_m$, $\mathbb{W}_m$ sont les matrices orthogonales construites en appliquant simultanément $m$ itérations de l'algorithme d'Arnoldi étendu par blocs aux paires de matrices $(A, E)$ et $(B^T, F)$ respectivement. Dans ce cas, on a

$$\mathbb{T}_A = \mathbb{V}_m^T A \mathbb{V}_m, \quad \textbf{et} \quad \mathbb{T}_B = \mathbb{W}_m^T B \mathbb{W}_m.$$

Pour trouver la matrice $Y_m^{GA} \in \mathbb{R}^{2mr\times 2mr}$ nous utilisons la condition d'orthogonalité suivante

$$\mathcal{R}_m^{GA} := AX_m^{GA}B - X_m^{GA} + EF^T \perp_F \mathbb{L}_m(A, B),$$

où $\mathbb{L}_m(A, B)$ est le sous espace des matrices de la forme $\mathbb{V}_m Y_m \mathbb{W}_m^T$. La condition d'orthogonalité ci-dessus permet de donner l'équation suivante :

$$\mathbb{V}_m^T \mathcal{R}_m^{GA} \mathbb{W}_m = 0.$$

Comme les matrices $\mathbb{V}_m$ et $\mathbb{W}_m$ sont orthogonales, donc il est plus facile de monter que $Y_m^{GA} \in \mathbb{R}^{2mr\times 2mr}$ est la solution de l'équation de Stein de taille réduite

$$\mathbb{T}_A Y_m^{GA} \mathbb{T}_B - Y_m^{GA} + \widetilde{E}\widetilde{F}^T = 0, \tag{3.36}$$

où $\widetilde{E} = \mathbb{V}_m^T E$ et $\widetilde{F} = \mathbb{W}_m^T F$.

En supposant que $\lambda_i(\mathbb{T}_A).\lambda_j(\mathbb{T}_B) \neq 1$ pour toutes $i = 1, 2, ..., 2mr$ et $j = 1, 2, ..., 2ms$, pour assurer l'existence et l'unicité de la solution $Y_m^{GA}$. La résolution de l'équation de Stein de taille réduite (3.36) peut être obtenue par une méthode directe, par exemple la méthode Hessenberg-Schur [33]. Pour tester la convergence, Il est nécessaire de calculer la norme de Frobenius du résidu $\mathcal{R}_m^{GA}$, pour cela le résultat suivant donne une expression du résidu $\mathcal{R}_m^{GA}$ en fonction des matrices de petite taille.

**Théorème 3.8.1.** *[63] Soient $Y_m^{GA}$ la solution exacte de l'équation de Stein réduite (3.36) et $X_m^{GA} = \mathbb{V}_m Y_m^{GA} \mathbb{W}_m^T$ la solution approchée de l'équation (3.32). Alors la norme de Frobenius du résidu $\mathcal{R}_m$ est donnée par*

$$\|\mathcal{R}_m^G(X_m^{GA})\|_F = \sqrt{\alpha_m^2 + \beta_m^2 + \gamma_m^2}, \tag{3.37}$$

*où les réels $\alpha_m$, $\beta_m$ et $\gamma_m$ sont définis par*

$$\alpha_m = \left\| \mathbb{T}_m^A Y_m \mathbb{E}_m (T_{m+1,m}^B)^T \right\|_F, \quad \beta_m = \left\| T_{m+1,m}^A \mathbb{E}_m^T (\mathbb{T}_m^B)^T \right\|_F,$$

*et*

$$\gamma_m = \left\| T_{m+1,m}^A \mathbb{E}_m^T Y_m \mathbb{E}_m (T_{m+1,m}^B)^T \right\|_F.$$

*Démonstration.* Soient $E = V_1 R_E$ et $F = W_1 R_F$ sont les décomposition $QR$ de $E$ et $F$ respectivement. D'après les relations (3.9) et (3.10) nous avons

$$\begin{aligned}
\mathcal{R}_m^{GA} &= \left\| A\mathbb{V}_m Y_m \mathbb{W}_m^T B - \mathbb{V}_m Y_m \mathbb{W}_m^T + EF^T \right\|_F \\
&= \left\| \mathbb{V}_{m+1} \bar{\mathbb{T}}_m^A Y_m \bar{\mathbb{T}}_m^{BT} \mathbb{W}_{m+1}^T - \mathbb{V}_m Y_m \mathbb{W}_m^T + V_1 R_E R_F^T W_1^T \right\|_F \\
&= \left\| \mathbb{V}_{m+1} \bar{\mathbb{T}}_m^A Y_m \bar{\mathbb{T}}_m^{BT} \mathbb{W}_{m+1}^T - \mathbb{V}_{m+1} \begin{pmatrix} I \\ 0 \end{pmatrix} Y_m \begin{pmatrix} I & 0 \end{pmatrix} \mathbb{W}_{m+1}^T + V_1 R_E R_F^T W_1^T \right\|_F \\
&= \left\| \mathbb{V}_{m+1} \left[ \bar{\mathbb{T}}_m^A Y_m \bar{\mathbb{T}}_m^{BT} - \begin{pmatrix} I \\ 0 \end{pmatrix} Y_m \begin{pmatrix} I & 0 \end{pmatrix} + \begin{pmatrix} R_E R_F^T & 0 \\ 0 & 0 \end{pmatrix} \right] \mathbb{W}_{m+1}^T \right\|_F
\end{aligned}$$

Comme les matrices $\mathbb{V}_{m+1}$ et $\mathbb{W}_{m+1}$ sont orthogonales, et

$$\overline{\mathbb{T}}_m^A = \begin{bmatrix} \mathbb{T}_m^A \\ T_{m+1,m}^A \mathbb{E}_m^T \end{bmatrix}, \quad \textbf{et} \quad \overline{\mathbb{T}}_m^B = \begin{bmatrix} \mathbb{T}_m^B \\ T_{m+1,m}^B \mathbb{E}_m^T \end{bmatrix}$$

Donc, nous avons

$$\begin{aligned}
\mathcal{R}_m^{GA} &= \left\| \left[ \begin{bmatrix} \mathbb{T}_m^A \\ T_{m+1,m}^A \mathbb{E}_m^T \end{bmatrix} Y_m \begin{bmatrix} \mathbb{T}_m^B \\ T_{m+1,m}^B \mathbb{E}_m^T \end{bmatrix}^T - \begin{pmatrix} Y_m & 0 \\ 0 & 0 \end{pmatrix} + \begin{pmatrix} R_E R_F^T & 0 \\ 0 & 0 \end{pmatrix} \right] \right\|_F \\
&= \left\| \left[ \begin{pmatrix} \mathbb{T}_m^A Y_m (\mathbb{T}_m^B)^T & \mathbb{T}_m^A Y_m \mathbb{E}_m (T_{m+1,m}^B)^T \\ T_{m+1,m}^A \mathbb{E}_m^T (\mathbb{T}_m^B)^T & T_{m+1,m}^A \mathbb{E}_m^T Y_m \mathbb{E}_m (T_{m+1,m}^B)^T \end{pmatrix} + \begin{pmatrix} -Y_m + R_E R_F^T & 0 \\ 0 & 0 \end{pmatrix} \right] \right\|_F \\
&= \left\| \begin{pmatrix} \mathbb{T}_m^A Y_m (\mathbb{T}_m^B)^T - Y_m + R_E R_F^T & \mathbb{T}_m^A Y_m \mathbb{E}_m (T_{m+1,m}^B)^T \\ T_{m+1,m}^A \mathbb{E}_m^T (\mathbb{T}_m^B)^T & T_{m+1,m}^A \mathbb{E}_m^T Y_m \mathbb{E}_m (T_{m+1,m}^B)^T \end{pmatrix} \right\|_F
\end{aligned}$$

Puisque $Y_m^{GA}$ la solution exacte de l'équation de Stein (3.36) donc nous avons

$$\mathcal{R}_m^{GA} = \left\| \begin{pmatrix} 0 & \mathbb{T}_m^A Y_m \mathbb{E}_m (T_{m+1,m}^B)^T \\ T_{m+1,m}^A \mathbb{E}_m^T (\mathbb{T}_m^B)^T & T_{m+1,m}^A \mathbb{E}_m^T Y_m \mathbb{E}_m (T_{m+1,m}^B)^T \end{pmatrix} \right\|_F.$$

Par conséquent

$$\left\| \mathcal{R}_m^{GA} \right\|_F^2 = \left\| \mathbb{T}_m^A Y_m \mathbb{E}_m (T_{m+1,m}^B)^T \right\|_F^2 + \left\| T_{m+1,m}^A \mathbb{E}_m^T (\mathbb{T}_m^B)^T \right\|_F^2 + \left\| T_{m+1,m}^A \mathbb{E}_m^T Y_m \mathbb{E}_m (T_{m+1,m}^B)^T \right\|_F^2.$$

Nous avons finalement le résultat (3.37) et nous rappelons que $\mathbb{E}_m^T = \begin{bmatrix} 0_{2r \times 2(m-1)r} & I_{2r} \end{bmatrix}$.

□

Ce résultat est très important puisqu'il nous permet de calculer la norme de Frobenius du résidu $\mathcal{R}_m(X_m^{GA})$ sans avoir a effectuer le produit des matrices de grande taille.

La méthode ainsi obtenue sera notée GA-N-Stein, et algorithme correspondant peut être formulé comme suit :

---

**Algorithme 3 : pour résoudre l'équation de Stein non symétrique (GA-Lr-SE)**

---

1. On choisit une tolérance $\epsilon > 0$ et un nombre maximal d'itérations $m_{max}$
2. **Pour** $m = 1, 2, 3, ..., m_{max}$
3. Construire les matrices $\mathbb{V}_m$ et $\mathbb{T}_m^A$ par l'algorithme 2 (Ch 1) appliqué au couple $(A, E)$.
4. Construire les matrices $\mathbb{W}_m$ et $\mathbb{T}_m^B$ par l'algorithme 2 (Ch 1) appliqué au couple $(B^T, F)$.
5. Résoudre l'équation de Stein projetée suivante

$$\mathbb{T}_m^A Y_m^{GA} (\mathbb{T}_m^B)^T - Y_m^{GA} + \widetilde{E}\widetilde{F}^T = 0$$

6. Calculer la norme de Frobenius du résidu :

$$\|\mathcal{R}_m^G(X_m^{GA})\|_F = \sqrt{\alpha_m^2 + \beta_m^2 + \gamma_m^2},$$

où

$$\alpha_m = \left\|\mathbb{T}_m^A Y_m \mathbb{E}_m (T_{m+1,m}^B)^T\right\|_F, \quad \beta_m = \left\|T_{m+1,m}^A \mathbb{E}_m^T (\mathbb{T}_m^B)^T\right\|_F,$$

et

$$\gamma_m = \left\|T_{m+1,m}^A \mathbb{E}_m^T Y_m \mathbb{E}_m (T_{m+1,m}^B)^T\right\|_F.$$

– **Si** $\|\mathcal{R}_m\| < \epsilon$ , **Aller à 8**
– **Sinon** $m = m + 1$
– **Fin si**

7. **Fin pour** $m$
8. Calculer la décomposition en valeurs singulières (SVD) de la matrice $Y_m^{GA}$, c'est à dire $Y_m^{GA} = \mathcal{V}\Sigma\mathcal{W}^T$, où $\Sigma = diag(\sigma_1, \sigma_2, ..., \sigma_{2mr})$ et $\sigma_1 \geq \sigma_2 \geq ... \geq \sigma_{2mr}$ ;
trouver $l$ telle que $\sigma_{l+1} \leq tol_{trn} < \sigma_l$ et soit $\Sigma_l = diag(\sigma_1, \sigma_2, ..., \sigma_l)$ ;
$Z_m^A = \mathbb{V}_m \mathcal{V}_l \Sigma_l^{1/2}$, et $Z_m^B = \mathbb{W}_m \mathcal{W}_l \Sigma_l^{1/2}$
9. La solution approchée $X_m$ est donnée par $X_m \approx Z_m^A \, Z_m^{BT}$.

---

L'équation de Stein projetée est résolu par les méthode directes.

### 3.8.2 Méthode de minimisation du résidu

Dans ce paragraphe, nous appliquons la méthode de minimisation de résidu (MR) à l'équation de Stein non symétrique. A chaque itération $m$ de l'algorithme d'Arnoldi étendu par blocs, on a une approximation $X_m^{MR}$ de l'équation de Stein non symétrique (3.32), où $X_m^{MR}$ la solution du problème de minimisation suivant :

$$X^{MR} = \arg \min_{X=\mathbb{V}_m Y_m \mathbb{W}_m^T} \left\| AXB - X + EF^T \right\|_F, \tag{3.38}$$

où $\mathbb{V}_m$, $\mathbb{W}_m \in \mathbb{R}^{n\times mr}$ sont les matrices orthogonales construites en appliquant simultanément $m$ itérations de l'algorithme d'Arnoldi étendu par blocs aux paires de matrices $(A, E)$ et $(B^T, F)$ respectivement. Donc, nous avons les relations suivantes :

$$A\mathbb{V}_m = \mathbb{V}_{m+1}\bar{\mathbb{T}}_m^A \tag{3.39}$$

et

$$B^T\mathbb{W}_m = \mathbb{W}_{m+1}\bar{\mathbb{T}}_m^B. \tag{3.40}$$

Nous avons le résultat suivant

**Théorème 3.8.2.** *Le problème de minimisation*

$$X_m^{MR} = \arg \min_{X_m=\mathbb{V}_m Y_m \mathbb{W}_m^T} \left\| AX_mB - X_m + EF^T \right\|_F$$

*est équivalent au problème de minimisation réduit suivant*

$$Y_m^{MR} = \arg\min_{Y_m} \left\| \left[ \bar{\mathbb{T}}_m^A Y_m (\bar{\mathbb{T}}_m^B)^T - \begin{bmatrix} I \\ 0 \end{bmatrix} Y_m \begin{bmatrix} I & 0 \end{bmatrix} + \begin{bmatrix} R_E R_F^T & 0 \\ 0 & 0 \end{bmatrix} \right] \right\|_F. \tag{3.41}$$

*où $E = V_1 R_E$ et $F = W_1 R_F$ sont les décomposition $QR$ de $E$ et $F$ respectivement.*

*Démonstration.* Soient $E = V_1 R_E$ et $F = W_1 R_F$ sont les décomposition $QR$ de $E$ et $F$ respectivement. D'après les relations (3.39) et (3.40) nous avons

$$\begin{aligned}
&\min_{X=\mathbb{V}_m Y_m \mathbb{W}_m^T} \left\| AXB - X + EF^T \right\|_F \\
&= \min_{Y_m} \left\| A\mathbb{V}_m Y_m \mathbb{W}_m^T B - \mathbb{V}_m Y_m \mathbb{W}_m^T + V_1 R_E R_F^T W_1^T \right\|_F \\
&= \min_{Y_m} \left\| \mathbb{V}_{m+1}\bar{\mathbb{T}}_m^A Y_m \bar{\mathbb{T}}_m^{BT} \mathbb{W}_{m+1}^T - \mathbb{V}_{m+1} \begin{pmatrix} I \\ 0 \end{pmatrix} Y_m \begin{pmatrix} I & 0 \end{pmatrix} \mathbb{W}_{m+1}^T + V_1 R_E R_F^T W_1^T \right\|_F \\
&= \min_{Y_m} \left\| \mathbb{V}_{m+1} \left[ \bar{\mathbb{T}}_m^A Y_m \bar{\mathbb{T}}_m^{BT} - \begin{pmatrix} I \\ 0 \end{pmatrix} Y_m \begin{pmatrix} I & 0 \end{pmatrix} + \begin{pmatrix} R_E R_F^T & 0 \\ 0 & 0 \end{pmatrix} \right] \mathbb{W}_{m+1}^T \right\|_F \\
&= \min_{Y_m} \left\| \left[ \bar{\mathbb{T}}_m^A Y_m \bar{\mathbb{T}}_m^{BT} - \begin{pmatrix} I \\ 0 \end{pmatrix} Y_m \begin{pmatrix} I & 0 \end{pmatrix} + \begin{pmatrix} R_E R_F^T & 0 \\ 0 & 0 \end{pmatrix} \right] \right\|_F.
\end{aligned}$$

□

Le problème de minimisation réduit (3.41) est résolu par les méthodes directes ou itératives comme le cas du problème de minimisation associé à l'équation de Sylvester.

On peut énoncer l'algorithme MR pour résoudre l'équation de Stein non symétrique comme suit

**Algorithm 4** Algorithme de MR pour résoudre l'équation de Stein non symétrique

- On choisit une tolérance $tol > 0$ et un nombre maximum d'itérations $iter_{max}$.
- Pour $m = 1, 2, 3, ..., iter_{max}$
- Construire les $\mathbb{V}_m, \overline{\mathbb{T}}_m^A$, exécuter l'algorithme EBA pour $(A, E)$.
- Construire les $\mathbb{W}_m, \overline{\mathbb{T}}_m^B$, exécuter l'algorithme EBA pour $(B^T F)$.
- Résoudre le problème de minimisation

$$Y_m^{MR} = \arg\min_{Y_m} \left\| \bar{\mathbb{T}}_m^A Y_m (\bar{\mathbb{T}}_m^B)^T - \begin{pmatrix} I \\ 0 \end{pmatrix} Y_m \begin{pmatrix} I & 0 \end{pmatrix} + \begin{pmatrix} R_E R_F^T & 0 \\ 0 & 0 \end{pmatrix} \right\|_F .$$

- Si $\|\mathcal{R}_m\|_F \leq tol$, Arrêt,
- La solution $X_m$ est donnée par $X_m \approx Z_{1,m}\, Z_{2,m}^T$.

## 3.9 Exemples numériques

Dans cette section, nous présentons quelques exemples numériques de l'équation matricielle de Sylvester et l'équation matricielle de Stein non Symétrique. Nous allons comparer l'approche de Galerkin (GA) avec notre approche : minimisation de résidu (MR), pour des problèmes de grande taille. En utilisant : la méthode LSQR globale (Gl-LSQR), la méthode du gradient conjugué globale préconditionné (PGCG), direct (le produit de Kronecker), et la méthode Hu et Reichel (HR), pour la résolution le problème de minimisation réduit. Les algorithmes ont été codés avec Matlab 2009. A chaque itération $m$ de l'algorithme d'Arnoldi étendu par blocs, équation matricielle de Sylvester projetée de taille $2mr \times 2mr$ a été résolue par la méthode Bartels-Stewart [11] pour l'approche de Galerkin (GA). Le problème de minimisation réduit est résolue par LSQR global (Gl-LSQR) et la méthode du gradient conjugué globale préconditionné (PGCG). On s'arrête si la norme de Frobenius du résidu est inférieure à $tol_l = 10^{-12}$ ou quand le nombre maximum d'itérations $kmax = 1000$ est atteint. Pour l'algorithme minimisation de résidu (MR) et l'algorithme de Galerkin s'arrête lorsque la norme de Frobenius du résidu est inférieure à une certaine tolérance $tol$ (qui sera précisée pour chaque figure) ou bien le nombre maximum d'itérations $itermax = 50$. Dans tous les tests, les coefficients

des matrices $E$ et $F$ ont été générés par des valeurs aléatoires uniformément distribuées dans l'intervalle $[0,\ 1]$. Les matrices $A$ et $B$ sont obtenues par à partir de la discrétisation par différences finies de l'opérateur suivant

$$L_u = \Delta u - f_1(x,y)\frac{\partial u}{\partial x} + f_2(x,y)\frac{\partial u}{\partial y} + g(x,y), \tag{3.42}$$

dans le domaine $[0,1] \times [0,1]$ avec les condition de Dirichlet homogène. La taille de la matrice construite est $n = n_0^2$, où $n_0$ est le nombre de points de la grille dans chaque direction. La discrétisation de l'opérateur $L_u$ donne des matrices dans la bibliothèque `Lyapack` [89] et désignés par

$$\text{A = fdm2d\_matrix(n0,'f\_1(x,y)','f\_2(x,y)','g(x,y)').}$$

On utilisera aussi des matrices de la collection Matrix-Market et aussi la matrice 'flow_meter' de la collection Oberwolfach (imtek.de/simulation/benchmark).

Les figures qui apparaissent dans la suite, nous donnons les courbes représentant la norme de Frobenius du résidu dans une échelle logarithmique en fonction du nombre d'itérations $m$. La courbe de la méthode de minimisation de résidu (MR) est représentée par ligne pointillée (verte) et celle de Galerkin (GA) par linge continue (en bleu). On donne également des tableaux pour comparer les méthodes proposées. Pour cela, on a fixé la tolérance à $tol = 10^{-7}$ pour les deux méthodes, et nous donnons : la norme de Frobenius du résidu, le nombre d'itérations ainsi que le temps CPU en secondes.

### 3.9.1 Tests numérique pour l'équation de Sylvester

Dans ce paragraphe, nous présentons quelque exemples numériques de l'équation de Sylvester de grande talle.

**Exemple 1.1** Dans ce premier exemple, les matrices $A$ et $B$ sont obtenus à partir de la discrétisation de l'opérateur $Lu$ et $L_B$ respectivement

$$\begin{aligned} L_A &= \Delta u - e^{xy}\,\frac{\partial u}{\partial x} - \sin(xy)\,\frac{\partial u}{\partial y} - (y^2 - x^2)\,u, \\ L_B &= \Delta u - 100\,e^{x}\frac{\partial u}{\partial x} - 10\,xy\,\frac{\partial u}{\partial y} - \sqrt{x^2+y^2}\,u, \end{aligned}$$

sur le carré unité $[0,1] \times [0,1]$ avec des conditions aux bord de Dirichlet homogènes. Les matrices $A$ et $B$ sont de taille $n = n_0^2$ et $s = s_0^2$ respectivement, où $n_0$ et $s_0$ sont les nombres de points de la grille intérieure dans la direction $x$ et $y$ respectivement.
Pour ce exemple, nous avons choisi $n = 4900$, $s = 3600$ et $r = 2$. on obtient la figure 1.1 ci-dessus.

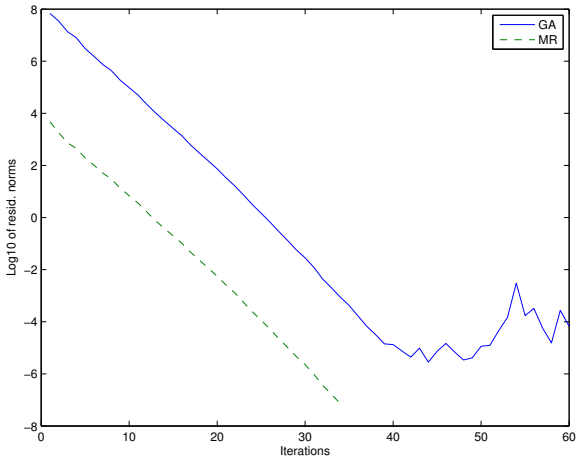


FIGURE 3.1: Résultat pour l'exemple 1.1.

Notons que le problème de minimisation réduit (associé à la méthode de minimisation du résidu MR) est résolu par la méthode itérative LSQR globale (Gl-LSQR).

**Exemple 1.2** Dans cet exemple, nous avons utilisé les matrices $A = pde2961$ et $B = thermal$ de la collection Harwell-Boeing (http ://math.nist.gov/MatrixMarket). Les matrices $A$ et $B$ sont de taille $n = 2961$ et $s = 3456$ respectivement. Pour ce test, nous avons choisi $r = 2$ et le problème de minimisation réduit a été résolu par la méthode direct des moindres carrés basé sur le produit de Kronecker (dans la section 3.4.1).

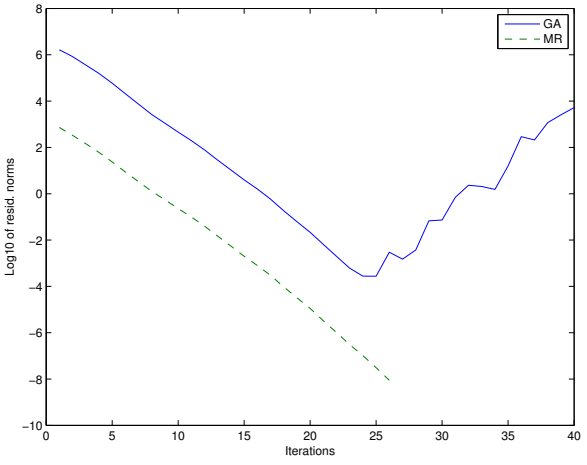


FIGURE 3.2: Résultat pour l'exemple 1.2.

**Exemple 1.3** Dans cet exemple, nous avons utilisé les matrices $A$ et $B$ sont obtenus à partir de la discrétisation de l'opérateur $Lu.$ avec des dimensions $n = 8100$ et $s = 3456$ respectivement. Dans la figure 3.3, nous utilisons les matrices

$$A = \text{fdm2d\_matrix}(90,'cos(x+y)','sin(y^2)','100'),$$

et $B = mmread('thermal.mtx')$; $r = 5$ et $tol = 10^{-10}$. Le problème de minimisation de taille réduit est résolu par la méthode de Hu et Reichel.

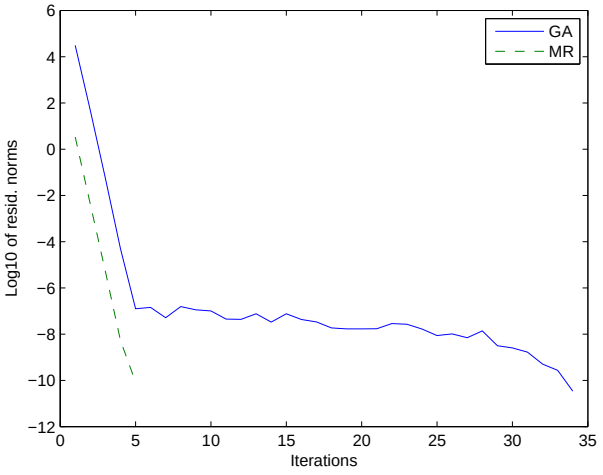


FIGURE 3.3: Résultat pour l'exemple 1.3.

**Exemple 1.4** Dans La figure 3.4. les matrices $A$ et $B$ sont obtenus à partir de la discrétisation de l'opérateur $Lu$ et $L_B$ respectivement, avec des dimensions $n = 4900$ et $s = 3600$, respectivement. Dans la figure 3.4, nous utilisons les matrices $A = add32$ ; $n = 4960$ et $B = \text{fdm2d\_matrix}(p_0,'x+y','x \times y','100')$; $r = 4$ et $tol = 10^{-10}$. Le problème de minimisation projeté résolu par la méthode Gl-LSQR.

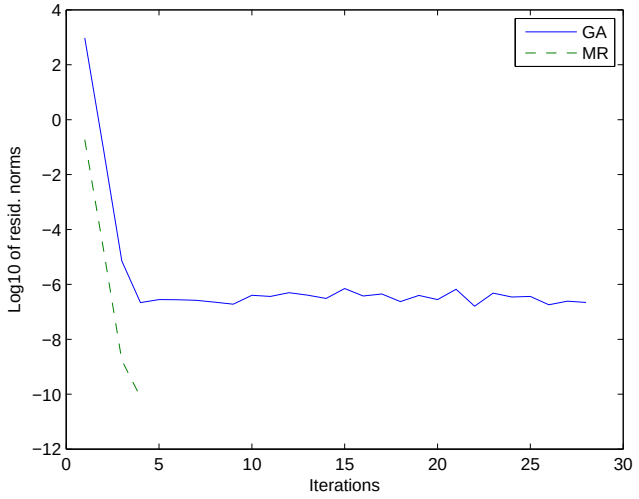


FIGURE 3.4: Résultat pour l'exemple 1.4.

**Exemple 1.5.** Dans cet exemple, nous considérons l'équation particulier de Sylvester suivante :

$$AX + XA = EF^T \tag{3.43}$$

Cette équation intervienne dans le calcul du grammien croisé d'un système dynamique linéaire invariant par rapport au temps. Pour plus de détails sur cette équation et la connexion avec les problèmes de réduction de modèle [5]. On choisi $n = s = 6400$, $r = 3$ et $tol = 10^{-9}$. Le problème de minimisation projeté résolu par la méthode PGCG.

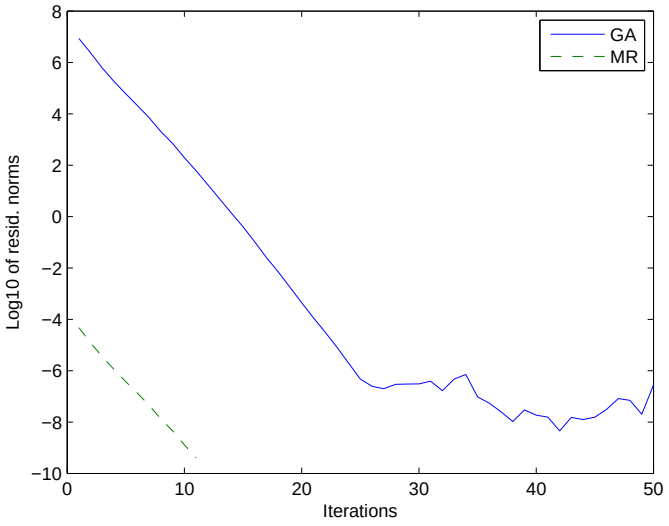


FIGURE 3.5: Résultat pour l'exemple 1.5.

Les exemples numériques confirment l'efficacité de la méthode proposée par rapport à l'approche de Galerkin.

**Exemple 1.6.** Dans cet exemple, nous avons comparé les performances de la méthode de minimisation du résidu MR et l'approche Galerkin. Le problème de minimisation réduit a été résolu par la méthode des moindres carrés direct, par LSQR globale (Gl-LSQR), par le gradient conjugué globale préconditionné (GPCG) ou par la méthode de Hu-Reichel (HR). Dans le tableau 3.1 suivant, nous donnons les résultats obtenus par différentes méthodes de MR et la méthode de Galerkin.
Les matrices $A$ et $B$ sont issues de la collection Matrix-Market collection et aussi la matrice 'flow_meter' de la collection Oberwolfach (imtek.de/simulation/benchmark), ou bien provenant de la discrétisation de l'opérateur $L_u$ (3.42). Pour ce test aussi, nous avons choisi $r = 2$.

D'après les résultats du tableau 3.1 que les méthode MR(GPCG) et MR(direct) sont plus rapides que les autre méthodes, et on observe aussi que la méthode MR(GPCG) donne les meilleurs résultats en terme de temps de calcul.

| **Les matrices** | **Méthode** | **Temps** | **N.d'itér** | $\parallel R_m \parallel_F$ |
|---|---|---|---|---|
| $A = thermal$, | MR(PGCG) | $26s$ | 10 | $1.1 \times 10^{-8}$ |
| $B = add32$ | MR (direct) | $26s$ | 9 | $2.1 \times 10^{-8}$ |
| $n = 3456$, $s = 4960$ | MR (Gl-LSQR) | $28s$ | 10 | $1.3 \times 10^{-8}$ |
| | MR(HR) | $27s$ | 9 | $3.5 \times 10^{-8}$ |
| | GA | $25s$ | 9 | $1.4 \times 10^{-8}$ |
| $A = fdm(cos(xy), e^{y^2 x}, 100)$ | MR(PGCG) | $45s$ | 15 | $1.9 \times 10^{-8}$ |
| $B = add32$ | MR (direct) | $56s$ | 15 | $2.3 \times 10^{-8}$ |
| $n = 90000$, $s = 4960$ | MR (Gl-LSQR) | $49s$ | 17 | $1.1 \times 10^{-8}$ |
| | MR(HR) | $88s$ | 15 | $3.1 \times 10^{-8}$ |
| | GA | $83s$ | 21 | $8.6 \times 10^{-5}$ |
| $A = fdm(sin(xy), e^{xy}, 10)$ | MR(PGCG) | $56s$ | 18 | $2.5 \times 10^{-8}$ |
| $B = thermal$ | MR(direct) | $58s$ | 16 | $3.1 \times 10^{-8}$ |
| $n = 122500$, $s = 3456$ | MR(Gl-LSQR) | $--$ | 50 | $----$ |
| | MR(HR) | $110s$ | 16 | $2.0 \times 10^{-8}$ |
| | GA | $82s$ | 25 | $8.4 \times 10^{-5}$ |
| $A = flow$ | MR (PGCG) | $49s$ | 16 | $2.0 \times 10^{-8}$ |
| $B = fdm(sin(xy), xy, 1000)$ | MR (direct) | $700s$ | 35 | $3.1 \times 10^{-8}$ |
| $n = 9669$, $s = 40000$ | MR(Gl-LSQR) | $--$ | 50 | $----$ |
| | MR(HR) | $150s$ | 35 | $2.4 \times 10^{-8}$ |
| | GA | $95s$ | 50 | $2.5 \times 10^{-5}$ |
| $A = fdm(xy, y^2, 1)$ | MR(PGCG) | $706s$ | 18 | $2.1 \times 10^{-8}$ |
| $B = fdm(xy, cos(xy), 10)$ | MR(direct) | $1500s$ | 42 | $1.5 \times 10^{-8}$ |
| $n = 122500$, $s = 48400$ | MR(Gl-LSQR) | $--$ | $--$ | $----$ |
| | MR(HR) | $--$ | $--$ | $----$ |
| | GA | $805s$ | 50 | $4.2 \times 10^{-4}$ |

TABLE 3.1: Résultats pour l'exemple 1.6

Maintenant, on va comparer la méthode MR(GPCG) avec la méthode ADI (Lr_ADI) [21, 23].

**Exemple 1.7**

Dans cet exemple, nous comparons notre méthode MR(GPCG) avec deux autres méthodes qui font référence dans la littérature : l'approche de GA et la méthode de rang inférieur ADI (Lr_ADI)[21, 23]. Pour cette expérience, les matrices $A$ et $B$ sont les mêmes que celles données dans [Exemple 1, [23]]. Elles sont obtenues à partir la discrétisation de l'opérateur $L_u$ (donnée par la relation (3.42)) à 5 points sur un maillage régulier. Les matrices $A$ et $B$ sont de taille $n = 6400$ et $s = 3600$ respectivement. Pour cette expérience, nous avons choisi $r = 4$ et la tolérance est $tol = 10^{-11}$.

| Méthode | Res. Norm. | Temps |
|---|---|---|
| MR(PGCG) | $1.2 \times 10^{-12}$ | $2.8s$ |
| GA | $6.4 \times 10^{-12}$ | $3.2s$ |
| Lr_ADI | $2.5 \times 10^{-12}$ | $5.2s$ |

TABLE 3.2: Résultats de l'exemple 1.7.

### 3.9.2 Tests numérique pour l'équation de Stein non symétrique

Dans ce paragraphe, nous présentons quelque exemples numériques de l'équation de Stein non symétrique de grande talle. Le problème de minimisation réduit associer à la méthode MR sera résolu soit par la méthode direct de Kronecker (direct) ou bien par la méthode du gradient conjugué globale préconditionné (PGCG). L'équation de Stein projetée (3.41) à été résolue en utilisant la fonction "dlyap" de Matlab. Les matrices $A$ et $B$ sont obtenus à partir de la discrétisation de l'opérateur $L_u$ (3.42), ou bien de la collection "Matrix-Market". Les matrices $E$ et $F$ sont des valeurs aléatoires uniformément distribuées dans l'intervalle $[0,\ 1]$.

**Exemple 2.1** Dans cet exemple, nous avons choisi

$$A = \text{fdm2d\_matrix}(90,'cos(x+y)','sin(y^2)','100'),$$

et $B =' thermal.mtx'$, donc les matrices $A$ et $B$ sont de taille $n = 8100$ et $s = 3456$ respectivement. On choisit aussi $r = 3$ et $tol = 10^{-9}$.
On obtient, pour les deux méthodes : MR(direct) et GA, les graphes suivants :

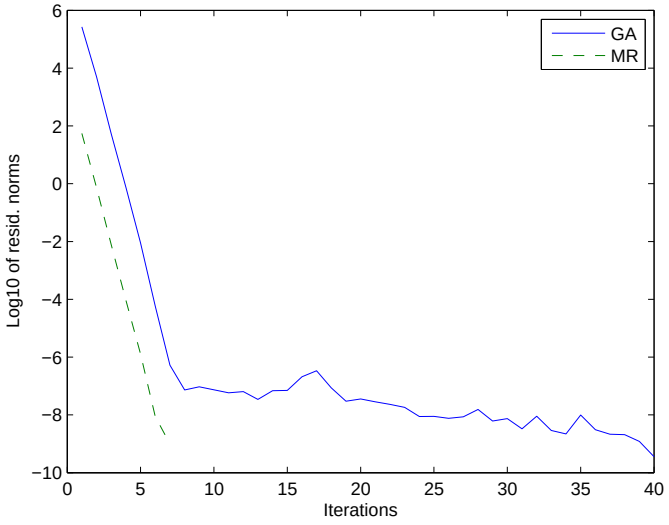


FIGURE 3.6: Résultats de l'exemple 2.1

**Exemple 2.2** On prend dans cet exemple, $A = flow$ et $B = thermal$, donc les matrices $A$ et $B$ sont de taille $n = 9669$ et $s = 3456$ respectivement. On prend aussi $r = 3$ et $tol = 10^{-9}$, et on obtient, pour la méthode MR(PGCG) et l'approche de Galrkin (GA), les courbes suivantes :

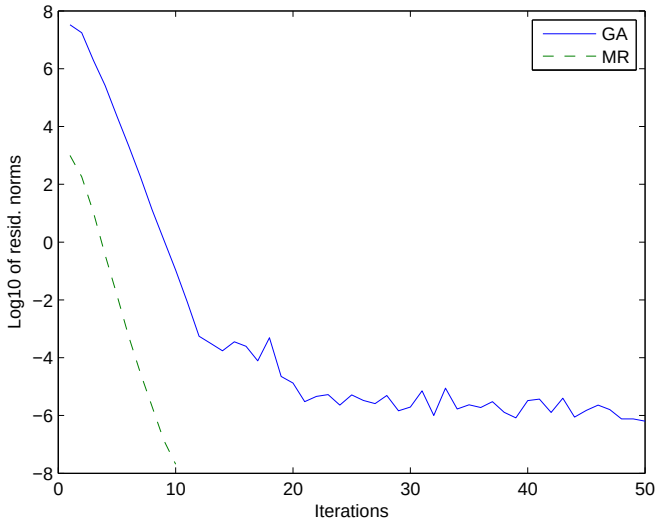


FIGURE 3.7: Résultats de l'exemple 2.2

**Exemple 2.3** Dans cet exemple, on augmente la taille de l'équation de Stein non Symétrique $n = 10000$ et $s = 8100$. Nous considérons $A$ = fdm2d_matrix(100,'$cos(x + y)$','$sin(y^2)$','100'), $B$ = fdm2d_matrix(90,'$x + y$','$xy$','200'). Pour $r = 4$ et $tol = 10^{-7}$, nous obtenons les courbes suivantes :

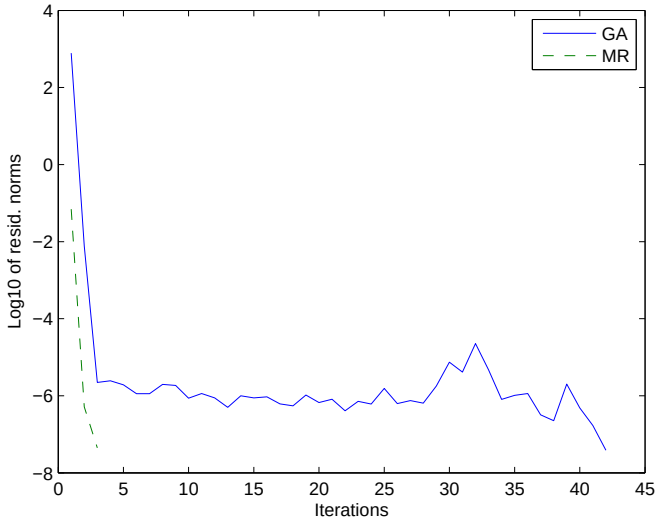


FIGURE 3.8: Résultats de l'exemple 2.3

**Exemple 2.4**. Dans cet exemple, on compare les méthodes trois méthodes MR(PGCG), MR(direct) et GA pour différentes matrices $A$ et $B$. Pour les tests dans le tableau (3.3), nous avons choisi $tol = 10^{-7}$ et $r = 2$. Dans le tableau (5.3), nous donnons les résultats obtenus par les trois méthodes.

| **Les matrices** | **Méthode** | **Temps** | **N.d'itér** | $\| \mathcal{R}_m \|_F$ |
|---|---|---|---|---|
| $A = thermal$, | MR(PGCG) | $55s$ | 4 | $5.05 \times 10^{-9}$ |
| $B = add32$ | MR (direct) | $58s$ | 6 | $1.22 \times 10^{-9}$ |
| $n = 3456$, $s = 4960$ | GA | $57s$ | 7 | $3.53 \times 10^{-8}$ |
| $A = fdm(cos(x+y), siny^2, 100)$ | MR(PGCG) | $\mathbf{5}s$ | 7 | $9.32 \times 10^{-8}$ |
| $B = fdm(x+y, yx, 200)$ | MR (direct) | $\mathbf{4}s$ | 3 | $3.03 \times 10^{-8}$ |
| $n = 10000$, $s = 6400$ | GA | $18s$ | 42 | $4.55 \times 10^{-8}$ |
| $A = fdm(sin(xy), e^{xy}, 10)$ | MR(PGCG) | $\mathbf{5}s$ | 3 | $7.29 \times 10^{-8}$ |
| $B = pde2961$ | MR(direct) | $128s$ | 18 | $9.10 \times 10^{-8}$ |
| $n = 14400$, $s = 3456$ | GA | $31s$ | 50 | $9.81 \times 10^{-7}$ |
| $A = flow$ | MR (PGCG) | **14s** | 14 | $6.65 \times 10^{-8}$ |
| $B = fdm(cos(x+y), sin(y^2), 1000)$ | MR (direct) | $178s$ | 20 | $4.06 \times 10^{-8}$ |
| $n = 9669$, $s = 12100$ | GA | $44s$ | 50 | $4.78 \times 10^{-5}$ |
| $A = fdm(e^{xy}, sin(xy), y^2 - x^2)$ | MR(PGCG) | **8s** | 9 | $4.59 \times 10^{-8}$ |
| $B = fdm(e^x, cos(xy), y^2)$ | MR(direct) | $230s$ | 26 | $1.78 \times 10^{-8}$ |
| $n = s = 10000$ | GA | $60s$ | 50 | $2.24 \times 10^{-2}$ |

TABLE 3.3: Résultats de l'exemple 2.4

Nous remarquons que la méthode MR(PGCG) permet de donner une bonne approximation, avec de meilleurs temps de calcul.

## 3.10 Conclusion

Dans ce chapitre, nous avons présenté une nouvelle méthode itérative pour résoudre l'équation de Sylvester ou Stein non symétrique de grande taille. La méthode proposée est basée sur la projection dans des sous-espaces de Krylov étendu (extended) par blocs avec la condition de minimisation du résidu, elle se ramène à la résolution d'un problème de minimisation réduit. Ce problème de minimisation réduit est résolu par des méthodes itératives ou directes. Pour accélérer la convergence de la résolution du problème de minimisation d'ordre inférieur, nous utilisons un préconditionnement pour la méthode de gradient conjugué global associé à l'équation normale. les approximations de la solution exacte sont données sous la forme factorisée : produit de deux matrices de rang inférieur

(low rank approximation), qui permet d'économiser de la mémoire pour les problèmes de grande dimension. L'avantage de la méthode de minimisation de résidu par rapport à l'approche de Galerkin est la stabilité numérique et également les meilleurs temps de calcul. Cela est démontré par les tests numériques effectués.

Chapitre 4

# Méthode de minimisation du résidu pour la résolution de l'équation de Riccati continue de grande taille

## 4.1 Introduction

Les équations de Riccati algébriques jouent un rôle fondamental dans de nombreux problèmes de la théorie du contrôle linéaire : contrôle linéaire avec coût quadratique, contrôle de type $H_\infty$ ou $H_2$, réduction de modèle, des équations différentielles etc, pour plus de détails voir [2, 3, 20, 33, 55, 64, 65, 80]. Le calcul de la solution de l'équation algébrique de Riccati fréquemment utilisée dans différents domaines comme par exemple les problèmes de calcul de contrôle linéaire quadratique (QLQ) suivant

$$\min \int_0^{+\infty} y(t)^T y(t) + u(t)^T u(t)\ dt \quad (QLQ) \tag{4.1}$$

où $y(t) \in \mathbb{R}^n$ et le contrôle $u(t) \in \mathbb{R}^n$ vérifient l'équation différentielle suivante :

$$\begin{cases} x'(t) = Ay(t) + Bu(t) \\ y(t) = Cx(t);\ x(0) = x_{0,} \end{cases} \tag{4.2}$$

avec $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times s}$, $C \in \mathbb{R}^{r\times n}$, $s << n$ et $r << n$.

Nous supposons que les matrices $B$ et $C$ sont de rang maximal $s$ et $r$ respectivement. Sous les conditions énoncées dans [109] : si le couple $(A,\ B)$ est c-stabilisable (c'est à dire qu'il existe une matrice $K$ telle que $A - BK$ soit stable) et $(C,\ A)$ est c-détectable (c'est à dire que $(A^T,\ C^T)$ est c-stabilisable) ; alors le problème (4.2) est minimisée par

$$u(t) = -B^T X x(t) \tag{4.3}$$

où $X$ est la solution symétrique semi-définie positive de l'équation de Riccati matricielle continue

$$A^T X + XA - XBB^T X + C^T C = 0. \tag{4.4}$$

Considérons la matrice Hamiltonienne $\mathcal{H}$ de dimension $2n \times 2n$ associée à l'équation (4.4)

$$\mathcal{H} = \begin{bmatrix} A & BB^T \\ C^T C & -A^T \end{bmatrix}$$

La solution de l'équation de Riccati est liée au calcul de sous espaces invariants de la matrice Hamiltonienne $\mathcal{H}$ associée. Si les matrices $Y, Z, W$ dans $\mathbb{R}^{n\times n}$, avec $Z$ non singulière, satisfont :

$$\mathcal{H}\begin{pmatrix} Y \\ Z \end{pmatrix} = \begin{pmatrix} Y \\ Z \end{pmatrix} W,$$

alors $X = YZ^{-1}$ est l'unique solution de l'équation algébrique de Riccati (4.4).
De nombreuses méthodes numériques ont été proposées pour résoudre l'équation de Riccati (4.4), on peut citer dans un premier temps, et pour les problèmes de petite taille l'approche par les valeurs propres consiste à calculer les sous-espaces invariants de la matrice Hamiltonienne $\mathcal{H}$ [16, 80, 91], comme les méthodes du type Lanczos symplectiques [16, 80, 107], ou la méthode de Newton [6, 14, 15, 80]. Pour les problèmes de grande taille, on a les méthodes de projection sur le sous-espaces de Krylov ou sur le sous-espace de Krylov par blocs [62, 65], ou bien sur le sous-espace de Krylov étendu (extended) par blocs [55]. Par exemple dans l'article de Heyouni et Jbilou [55], on cherche à construire une solution approchée $X_m^{GA}$ de la solution $X$ de l'équation de Riccati (4.4) sous la forme

$$X_m^{GA} = \mathbb{V}_m Y_m^{GA} \mathbb{V}_m^T.$$

dont la condition de Galerkin est définie comme suit :

$$\mathbb{V}_m^T (A^T X m^{GA} + X_m^{GA} A - X_m^{GA} BB^T X_m^{GA} + C^T C)\mathbb{V}_m = 0$$

Notre approche consiste à chercher la solution approchée $X_m^{MR}$ de la solution $X$ de l'équation de Riccati (4.4) sous la forme

$$X_m^{MR} = \mathbb{V}_m Y_m^{MR} \mathbb{V}_m^T$$

mais avec la condition de minimisation suivante

$$X^{MR} = \arg \min_{X=\mathbb{V}_m Y_m \mathbb{V}_m^T} \left\| AX + XA^T - XBB^T X + CC^T \right\|_F \tag{4.5}$$

où $\mathbb{V}_m$ est une matrice orthonormée.
Nous proposons dans ce travail une nouvelle méthode itérative basée sur la projection sur un sous-espace de Krylov étendu (extended) par blocs et la minimisation de résidu MR (Minimal Residual). Nous montrons, à travers des exemples numériques en fin de

ce chapitre, que la méthode proposée donne de bons résultats numériques comparés à la méthode de Galerkin [55]. Le reste du chapitre est organisé comme suit : Dans la Section 3, nous allons rappeler la méthode de Galerkin pour résoudre l'équation de Riccati de grande taille, cette méthode à été proposée par Heyouni et Jbilou [55]. Puis, la Section 4 introduit notre approche. La section 6 est consacrée à la méthode de MINRES globale pour résoudre une équation matricielle linéaire de type Lyapunov généralisée de petite taille. La Section 7, concernant la forme factorisée de la solution approchée, Puis, la Section 8 et 9 l'algorithme générale pour résoudre l'équation de Riccati symétrique de grande taille par l'approche de Galerkin et l'approche de minimisation de résidu, et on termine ce chapitre par les exemples numériques et une petite conclusion.

## 4.2 Méthode de Galerkin

Dans cette section, on va rappeler la méthode de projection sur le sous-espace de Krylov étendu par blocs avec la condition d'orthogonalité de Galerkin pour résoudre l'équation de Riccati de grande taille. Cette méthode est proposée par Heyouni et Jbilou en 2009 [55].

On considère l'équation de Riccati (4.4)

$$A^T X + XA - XBB^T X + C^T C = 0$$

On cherche à construire une suite d'approximations de petit rang $(X_m^{GA})_{m\in\mathbb{N}^*}$ de la solution $X$ sous la forme

$$X_m^{GA} = \mathbb{V}_m Y_m^{GA} \mathbb{V}_m^T \tag{4.6}$$

où $\mathbb{V}_m$ est la matrice orthonormale construite en appliquant l'algorithme d'Arnoldi étendu par blocs à la paire $(A^T, C^T)$. La matrice $Y_m^{GA} \in \mathbb{R}^{2mr\times 2mr}$ est déterminée en imposant la condition d'orthogonalité de Galerkin

$$\mathcal{R}_m(X_m) := A^T X_m^{GA} + X_m^{GA} A - X_m^{GA} BB^T X_m^{GA} + C^T C \perp_F \mathcal{L}_m(A^T, C^T).$$

où $\mathcal{L}_m(A^T, C^T)$ est le sous espace vectoriel de $\mathbb{R}^{n\times n}$ constitués des matrices de la forme $X = \mathbb{V}_m Y \mathbb{V}_m^T$.

c'est à dire

$$\mathbb{V}_m^T \mathcal{R}_m \mathbb{V}_m = 0.$$

Nous pouvons facilement montrer que $Y_m^{GA} \in \mathbb{R}^{2mr\times 2mr}$ est la solution de l'équation de Riccati de taille réduite

$$\mathbb{T}_A Y_m^{GA} + Y_m^{GA}\mathbb{T}_A^T - Y_m^{GA}\widetilde{B}\widetilde{B}^T Y_m^{GA} + \widetilde{C}\widetilde{C}^T = 0, \tag{4.7}$$

où $\mathbb{T}_A = \mathbb{V}_m^T A^T \mathbb{V}_m$, $\widetilde{C} = \mathbb{V}_m^T C^T$ et $\widetilde{B} = \mathbb{V}_m^T B_m$. La solution $Y_m^{GA}$ de l'équation de Riccati de taille réduite peut être obtenue par une méthode directe comme celles qui sont décrites dans [16, 27, 88]. Pour tester la convergence, il est nécessaire de calculer la norme du résidu $\mathcal{R}_m$, pour cela le résultat suivant donne une expression de la norme de Frobenius du résidu $\mathcal{R}_m$ en fonction des matrices de petite taille.

**Théorème 4.2.1.** *Soit $Y_m^{GA}$ la solution exacte de l'équation réduite (4.7) obtenue par l'algorithme d'Arnoldi étendu par blocs. On a alors la norme de Frobenius du résidu $\mathcal{R}_m := \mathcal{R}_m(X_m)$ est donnée par*

$$\| \mathcal{R}_m \|_F = \sqrt{2} \, \| \mathcal{T}_{m+1,m}^A \mathbb{E}_m^T Y_m^{GA} \|_F \tag{4.8}$$

*Démonstration.* La démonstration est donnée dans ([64]). □

L'intérêt pratique de ce résultat sur le résidu est de nous permettre de mettre en place un test d'arrêt sur l'algorithme sans avoir a calculer le produit $\mathcal{V}_m Y_m^{GA} \mathcal{V}_m^T$ à chaque itération.

Nous donnons ci-dessous une majoration de la norme de l'erreur $\|X - X_m^{GA}\|$, où $X$ est la solution exacte de l'équation de Riccati (4.4).

**Théorème 4.2.2.** *Soient $\tilde{Y}_m$ la matrice de taille $2s \times 2ms$ correspondante aux $2s$ dernières lignes de la matrice de $Y_m^{GA}$, $\gamma_m = \|T_{m+1,m}\tilde{Y}_m\|$, $\eta = \|BB^T\|$, $A_m = A - BB^T X_m^{GA}$ et supposons que $\sigma_m = sep(A_m, -A_m^T) > 0$. Alors si $4\gamma_m\eta/\delta_m^2 < 1$, nous avons*

$$\|X - X_m^{GA}\| \leq \frac{2\gamma_m}{\delta_m + \sqrt{\delta_m^2 - 4\gamma_m\eta}}. \tag{4.9}$$

*Démonstration.* La preuve est donnée dans ([64]). □

## 4.3 Méthode de minimisation du résidu

La plupart des méthodes proposées ces dernières années pour résoudre l'équation de Riccati continue de grande taille sont des méthodes de projection sur les sous-espaces de Krylov, par exemple : méthode de projection sur le sous espaces de Krylov par blocs [62] ou sur sous espaces de Krylov étendu par blocs [55]. Pour d'autres méthodes on peut citer [14, 22, 60, 64, 66, 98, 99]. Ces méthodes génèrent des approximations de petit rang de la solution de l'équation (4.4). Ces méthodes qui utilisent la condition de Galerkin,

présentent l'avantage de converger rapidement. Nous proposons une nouvelle méthode de type projection sur le sous espace de Krylov étendu avec la condition de minimisation du résidu à la place de la condition de Galerkin. L'objectif de cette méthode MR est de minimiser la norme du résidu sur $\mathcal{K}_m^e(A^T, C^T)$.

Comme les approximations de la solution exacte sont de la forme :

$$X_m = \mathbb{V}_m Y_m \mathbb{V}_m^T, \tag{4.10}$$

on va chercher à construire une approximation $X_m^{MR}$ à l'étape $m$, liée $\mathcal{K}_m^e(A^T, C^T)$. Cette approximation $X_m^{MR}$ est de la forme

$$X^{MR} = \mathbb{V}_m Y_m^{MR} \mathbb{V}_m^T,$$

et $X_m^{MR}$ est la solution du problème de minimisation suivant

$$X_m^{MR} = \arg \min_{X \in \mathbb{L}_m(A^T, C^T)} \left\| A^T X + XA - XBB^T X + C^T C \right\|_F, \tag{4.11}$$

où $\mathbb{L}_m(A^T, C^T)$ est l'ensemble des matrices de la forme $X = \mathbb{V}_m Y \mathbb{V}_m^T$, avec $\mathbb{V}_m = [V_1, V_2, \cdots, V_m]$ est la matrice orthogonale construite en appliquant l'algorithme d'Arnoldi étendu par blocs à $(A^T, C^T)$ et $m$ un entier naturel. Dans ce cas nous avons la relation suivante

$$A\mathbb{V}_m = \mathbb{V}_{m+1} \bar{\mathbb{T}}_m^A$$

Le théorème suivant permet de transformer le problème de minimisation grande taille (4.11) en un problème de minimisation de petite taille.

**Théorème 4.3.1.** *Soit $\mathbb{V}_m$ une matrice orthonormale construite en appliquant l'algorithme d'Arnoldi étendu par blocs, aux paire $(A^T, C^T)$. Alors le problème de minimisation*

$$X_m^{MR} = \arg \min_{X_m = \mathbb{V}_m Y_m \mathbb{V}_m^T} \left\| A^T X_m + X_m A - XBB^T X + C^T C \right\|_F$$

*est équivalent au problème de minimisation réduit suivant*

$$Y_m^{MR} = \arg\min_{Y_m} \left\| \bar{\mathbb{T}}_m Y_m \begin{bmatrix} I & 0 \end{bmatrix} + \begin{bmatrix} I \\ 0 \end{bmatrix} Y_m \bar{\mathbb{T}}_m^T - \begin{bmatrix} I \\ 0 \end{bmatrix} Y_m \tilde{B}\tilde{B}^T Y_m \begin{bmatrix} I & 0 \end{bmatrix} + \begin{bmatrix} R_C R_C^T & 0 \\ 0 & 0 \end{bmatrix} \right\|_F \tag{4.12}$$

*où $V_1 R_C$ est la décomposition $QR$ de $C^T$ et $\tilde{B} = \mathbb{V}^T B$.*

*Démonstration.* Soient $V_1 R_C$ la décomposition $QR$ de $C^T$, et $\mathbb{V}_m$ la matrice orthogonale construite en appliquant l'algorithme d'Arnoldi étendu par blocs à $(A^T, C^T)$. On a

$\mathbb{V}_m^T A^T = \mathbb{V}_{m+1}\bar{\mathbb{T}}_m^A$ et la matrice $\mathbb{V}_{m+1}$ est orthogonale. Ce qui nous donne,

$$\begin{aligned}
&\min_{X=\mathbb{V}_m Y_m \mathbb{V}_m^T} \left\| A^T X + XA - XBB^T X + CC^T \right\|_F \\
&= \min_{Y_m} \left\| A^T \mathbb{V}_m Y_m \mathbb{V}_m^T + \mathbb{V}_m Y_m \mathbb{V}_m^T A - \mathbb{V}_m Y_m \tilde{B}\tilde{B}^T \mathbb{V}_m Y_m \mathbb{V}_m^T + V_1 R_C R_C^T W_1^T \right\|_F \\
&= \min_{Y_m} \left\| \mathbb{V}_{m+1} \left[ \bar{\mathbb{T}}_m Y_m \begin{bmatrix} I & 0 \end{bmatrix} + \begin{bmatrix} I \\ 0 \end{bmatrix} Y_m \bar{\mathbb{T}}_m^T - \begin{bmatrix} I \\ 0 \end{bmatrix} Y_m \tilde{B}\tilde{B}^T Y_m \begin{bmatrix} I & 0 \end{bmatrix} + \begin{bmatrix} R_C R_C^T & 0 \\ 0 & 0 \end{bmatrix} \right] \mathbb{V}_{m+1}^T \right\|_F \\
&= \min_{Y_m} \left\| \left[ \bar{\mathbb{T}}_m Y_m \begin{pmatrix} I & 0 \end{pmatrix} + \begin{bmatrix} I \\ 0 \end{bmatrix} Y_m \bar{\mathbb{T}}_m^T - \begin{bmatrix} I \\ 0 \end{bmatrix} Y_m \tilde{B}\tilde{B}^T Y_m \begin{bmatrix} I & 0 \end{bmatrix} + \begin{bmatrix} R_C R_C^T & 0 \\ 0 & 0 \end{bmatrix} \right] \right\|_F .
\end{aligned}$$

où $\tilde{B} = \mathbb{V}^T B$. □

## 4.4 Résolution du problème de minimisation réduit

A chaque étape $m$ de l'algorithme d'Arnoldi étendu par blocs, nous déterminons une approximation de la solution $Y_m^{MR}$ du problème (4.11) que nous posons sous la forme,

$$\min_Y \|F(Y)\|_F \tag{4.13}$$

où $F$ est l'opérateur non linéaire

$$F(Y) = \mathcal{A}Y\mathcal{B} + \mathcal{B}^T Y \mathcal{A}^T + \mathcal{B}^T Y \mathcal{E} Y \mathcal{B} + \mathcal{F}.$$

Plus précisément

$$\mathcal{A} = \overline{\mathbb{T}}_m,\ \mathcal{E} = -\widetilde{B}\widetilde{B}^T,\ \mathcal{B} = \begin{bmatrix} I & 0 \end{bmatrix}\ et\ \mathcal{F} = \begin{bmatrix} R_C R_C^T & 0 \\ 0 & 0 \end{bmatrix}.$$

Le problème de minimisation (4.13) est équivalent au problème suivant

$$\min_Y f(Y), \tag{4.14}$$

où

$$f(Y) := \min_Y \|F(Y)\|_F^2 = < F(Y), F(Y) >_F .$$

Il est claire que $f$ est différentiable, et la différentielle de $f$ en $Y$ est :

$$df_Y(H) = \lim_{t \longrightarrow 0} \frac{f(Y+tH) - f(Y)}{t},\ \forall\ H \in \mathbb{R}^{n \times n}. \tag{4.15}$$

On a

$$F(Y) = \mathcal{A}Y\mathcal{B} + \mathcal{B}^T Y \mathcal{A}^T + \mathcal{B}^T Y \mathcal{E} Y \mathcal{B} + \mathcal{F}$$

Donc pour tout réel $t$ on a

$$F(Y+tH) = \mathcal{A}Y\mathcal{B}+t\mathcal{A}H\mathcal{B}+\mathcal{B}^TY\mathcal{A}^T+t\mathcal{B}^TH\mathcal{A}^T+\mathcal{B}^TY\mathcal{E}(Y+tH)\mathcal{B}+t\mathcal{B}^TH\mathcal{E}(Y+tH)\mathcal{B}+\mathcal{F}$$

$$= \mathcal{A}Y\mathcal{B}+\mathcal{B}^TY\mathcal{A}^T+\mathcal{F}+t(\mathcal{A}H\mathcal{B}+\mathcal{B}^TH\mathcal{A}^T)+\mathcal{B}^TY\mathcal{E}Y\mathcal{B}+t\mathcal{B}^TY\mathcal{E}H\mathcal{B}+t\mathcal{B}^TH\mathcal{E}Y\mathcal{B}+t^2\mathcal{B}^TH\mathcal{E}H\mathcal{B}$$

$$= F(Y) + t(\mathcal{A}H\mathcal{B} + \mathcal{B}^TH\mathcal{A}^T + \mathcal{B}^TY\mathcal{E}H\mathcal{B} + \mathcal{B}^TH\mathcal{E}Y\mathcal{B}) + t^2(\mathcal{B}^TH\mathcal{E}H\mathcal{B}).$$

Pour simplifier le calcul on pose

$$\Phi(H) = \mathcal{A}H\mathcal{B} + \mathcal{B}^TH\mathcal{A}^T + \mathcal{B}^TY\mathcal{E}H\mathcal{B} + \mathcal{B}^TH\mathcal{E}Y\mathcal{B}.$$

Donc

$$\begin{aligned} f(Y+tH) &= < F(Y+tH), F(Y+tH) >_F \\ &= < F(Y) + t\Phi(H) + t^2\mathcal{B}^TH\mathcal{E}H\mathcal{B}, F(Y) + t\Phi(H) + t^2\mathcal{B}^TH\mathcal{E}H\mathcal{B} >_F \\ &= < F(Y), F(Y) >_F + 2t < F(Y), \Phi(H) >_F + t^2(2 < F(Y), \mathcal{B}^TH\mathcal{E}H\mathcal{B} >_F \\ &+ < \Phi(H), \Phi(H) >_F) + 2t^3(< \mathcal{B}^TH\mathcal{E}H\mathcal{B}, \Phi(H) >_F) + t^4 \parallel \mathcal{B}^TH\mathcal{E}H\mathcal{B} \parallel_F \end{aligned}$$

Alors

$$\begin{aligned} df_Y(H) &= 2 < F(Y), \Phi(H) >_F \\ &= 2 < F(Y), \mathcal{A}H\mathcal{B} + \mathcal{B}^TH\mathcal{A}^T + \mathcal{B}^TY\mathcal{E}H\mathcal{B} + \mathcal{B}^TH\mathcal{E}Y\mathcal{B} >_F \\ &= 2 < \mathcal{A}^TF(Y)\mathcal{B}^T + \mathcal{B}F(Y)\mathcal{A} + \mathcal{E}^TY^T\mathcal{B}F(Y)\mathcal{B}^T + \mathcal{B}F(Y)\mathcal{B}^TY^T\mathcal{E}^T, H >_F \end{aligned}$$

Comme $Y$ est symétrique, nous avons

$$df_Y(H) = 2 < \mathcal{A}^TF(Y)\mathcal{B}^T + \mathcal{B}F(Y)\mathcal{A} + \mathcal{E}^TY\mathcal{B}F(Y)\mathcal{B}^T + \mathcal{B}F(Y)\mathcal{B}^TY\mathcal{E}^T, H >_F$$

La condition nécessaire de minimisation est que $df_Y = 0$ ce qui permet de donner l'équation suivante :

$$\forall H : \ 2 < \mathcal{A}^TF(Y)\mathcal{B}^T + \mathcal{B}F(Y)\mathcal{A} + \mathcal{E}^TY\mathcal{B}F(Y)\mathcal{B}^T + \mathcal{B}F(Y)\mathcal{B}^TY\mathcal{E}^T, H >_F = 0.$$

Ce qui est équivalent à :

$$\Psi(Y) := \mathcal{A}^TF(Y)\mathcal{B}^T + \mathcal{B}F(Y)\mathcal{A} + \mathcal{E}^TY\mathcal{B}F(Y)\mathcal{B}^T + \mathcal{B}F(Y)\mathcal{B}^TY\mathcal{E}^T = 0 \qquad (4.16)$$

Nous allons résoudre numériquement l'équation matricielle non linéaire (4.16) par la méthode de Newton, donc la dérivée de Fréchet de l'application non linéaire $\Psi(Y)$ en $Y$

est donnée par :

$$\begin{aligned}\Psi'_Y(Z) &= \mathcal{A}^T F'_Y(Z)\mathcal{B}^T + \mathcal{B}F'_Y(Z)\mathcal{A} + \mathcal{E}^T Z\mathcal{B}F(Y)\mathcal{B}^T + \mathcal{E}^T Y\mathcal{B}F'_Y(Z)\mathcal{B}^T \\ &+ \mathcal{B}F'_Y(Z)\mathcal{B}^T Y\mathcal{E}^T + \mathcal{B}F(Y)\mathcal{B}^T Z\mathcal{E}^T \\ &= (\mathcal{A}^T + \mathcal{E}^T Y\mathcal{B})F'_Y(Z)\mathcal{B}^T + \mathcal{B}F'_Y(Z)(\mathcal{A} + \mathcal{B}^T Y\mathcal{E}^T) + \mathcal{E}^T Z\mathcal{B}F(Y)\mathcal{B}^T \\ &+ \mathcal{B}F(Y)\mathcal{B}^T Z\mathcal{E}^T.\end{aligned}$$

où $F'_Y(Z) = \mathcal{B}^T Z(\mathcal{A}^T + \mathcal{E}Y\mathcal{B}) + (\mathcal{A} + \mathcal{B}^T Y\mathcal{E})ZB$.
On remplace $F'_Y(Z)$ dans $\Psi'_Y(Z)$ ce qui donne

$$\begin{aligned}\Psi'_Y(Z) &= (\mathcal{A}^T + \mathcal{E}^T Y\mathcal{B})\mathcal{B}^T Z(\mathcal{A}^T + \mathcal{E}Y\mathcal{B})\mathcal{B}^T + (\mathcal{A}^T + \mathcal{E}^T Y\mathcal{B})(\mathcal{A} + \mathcal{B}^T Y\mathcal{E})Z\mathcal{B}\mathcal{B}^T \\ &+ \mathcal{B}\mathcal{B}^T Z(\mathcal{A}^T + \mathcal{E}Y\mathcal{B})(\mathcal{A} + \mathcal{B}^T Y\mathcal{E}^T) + \mathcal{B}(\mathcal{A} + \mathcal{B}^T Y\mathcal{E})ZB(\mathcal{A} + \mathcal{B}^T Y\mathcal{E}^T) \\ &+ \mathcal{E}^T Z\mathcal{B}F(Y)\mathcal{B}^T + \mathcal{B}F(Y)\mathcal{B}^T Z\mathcal{E}^T\end{aligned}$$

Comme $\mathcal{B}\mathcal{B}^T = I$ et la matrice $\mathcal{E}$ symétrique donc nous avons

$$\begin{aligned}\Psi'_Y(Z) &= (\mathcal{A}^T\mathcal{B}^T + \mathcal{E}^T Y)Z(\mathcal{A}^T\mathcal{B}^T + \mathcal{E}^T Y) + (\mathcal{A}^T\mathcal{B}^T + \mathcal{E}^T Y)^T Z(\mathcal{A}^T\mathcal{B}^T + \mathcal{E}^T Y)^T \\ &+ (\mathcal{A}^T + \mathcal{E}^T Y\mathcal{B})(\mathcal{A} + \mathcal{B}^T Y\mathcal{E})Z + Z(\mathcal{A}^T + \mathcal{E}Y\mathcal{B})(\mathcal{A} + \mathcal{B}^T Y\mathcal{E}^T) \\ &+ \mathcal{E}Z(\mathcal{B}\mathcal{A}Y + Y\mathcal{A}^T\mathcal{B}^T + Y\mathcal{E}Y + \mathcal{B}\mathcal{F}\mathcal{B}^T) + (\mathcal{B}\mathcal{A}Y + Y\mathcal{A}^T\mathcal{B}^T + Y\mathcal{E}Y + \mathcal{B}\mathcal{F}\mathcal{B}^T)Z\mathcal{E}.\end{aligned}$$

Pour simplifier

$$\mathcal{M}(Z) := \Psi'_Y(Z) = A_1 Z A_1 + A_1^T Z A_1^T + A_2 Z + Z A_2 + \mathcal{E}Z A_3 + A_3 Z\mathcal{E} \tag{4.17}$$

où $A_1 = \mathcal{A}^T\mathcal{B}^T + \mathcal{E}Y$, $A_2 = (\mathcal{A}^T + \mathcal{E}Y\mathcal{B})(\mathcal{A} + \mathcal{B}^T Y\mathcal{E})$,

$$A_3 = \mathcal{B}\mathcal{A}Y + Y\mathcal{A}^T\mathcal{B}^T + Y\mathcal{E}Y + \mathcal{B}\mathcal{F}\mathcal{B}^T.$$

On va construire une suite de matrices $Y_k$ d'approximations de la solution exacte $Y$ de l'équation $\Psi(Y) = 0$. On choisit un premier terme $Y_0 = 0$ et, en supposant $Y_k$ connu, on définit $Y_{k+1}$ comme suit

$$Y_{k+1} = Y_k + N_k$$

où la matrice $N_k$ est solution de l'équation linéaire matricielle

$$\Psi'_{Y_k}(N_k) = -\Psi(Y_k) \tag{4.18}$$

**Remarque 4.4.1.** *L'opérateur linéaire $\mathcal{M}$ défini par (4.17) est symétrique par rapport au produit scalaire de Frobenius, c'est-à-dire :*

$$\forall, V \text{ et } W, \text{ on a } \langle \mathcal{M}(V), W\rangle_F = \langle V, \mathcal{M}(W)\rangle_F.$$

Pour résoudre l'équation matricielle linéaire symétrique (4.18), nous proposons la méthode MINRES globale.

## 4.5 Méthode de MINRES globale.

La méthode MINRES (MINimum RESidual method) de Paige et Saunders [ [94], SIAM J. Numer. Anal., 1975], est une méthode itérative très souvent utilisée pour résoudre des systèmes linéaires creux $Ax = b$ dans le cas $A$ est symétrique, en utilisant le processus Lanczos et la propriété de minimisation de la norme du résidu.

Dans cette section, nous proposons la méthode de MINRES globale (Gl-MINRES) pour résoudre les équations matricielles linéaires symétriques comme :

$$\mathcal{M}(Z) = \mathcal{C} \tag{4.19}$$

où $\mathcal{M} : \mathbb{R}^{n\times n} \longrightarrow \mathbb{R}^{n\times p}$ l'opérateur matriciel linéaire symétrique et $\mathcal{C}$ est une matrice de taille $n \times p$.

Soit $V$ une matrice réelle de taille $n \times p$ . Soit $k$ un entier naturel, on définit l'opérateur $\mathcal{M}^k$ par la relation de récurrence suivante

$$\mathcal{M}^0 = I \;\; et \;\; \mathcal{M}^k(V) = \mathcal{M}(\mathcal{M}^{k-1}(V)).$$

On définit le sous-espace de Krylov d'ordre $k$, associé à $\mathcal{M}$ et $V$ par :

$$\mathcal{K}_k(\mathcal{M}, V) = \text{sev}\{V, \mathcal{M}(V), ..., \mathcal{M}^{k-1}(V)\}$$

engendré par les matrices $V, \mathcal{M}(V), ..., \mathcal{M}^{k-1}(V)$, où $k$ est un entier naturel non nul. L'ensemble $\mathcal{K}_k(\mathcal{M}, V)$ ainsi défini est un sous-espace vectoriel de $\mathbb{R}^{n\times n}$.

L'algorithme de Lanczos global [70], dont les étapes sont données ci-dessous, produit une base orthonormale $\{V_1, ..., V_k\}$ de l'espace de Krylov $\mathcal{K}_k(\mathcal{M}, V)$ au sens de la norme de Frobenius, c'est à dire telle que $\langle V_i, V_j\rangle_F = \delta_{i,j},\ 1 \leq i, j \leq k$.

**Algorithm 5** L'algorithme de Lanczos symétrique global.

1. Initialisation : $V_1 = V/\|V\|_F$
2. Calculer $\tilde{V}_1 = \mathcal{M}(V_1)$
3. $\alpha_1 = \langle V_1, \tilde{V}_1 \rangle_F$
4. $\tilde{V}_1 = \tilde{V}_1 - \alpha_1 V_1$
5. $\beta_1 = \|\tilde{V}_1\|_F$
6. $V_2 = \tilde{V}_1/\beta_1$
7. Itérations :**Pour** $j = 2 : k$ **faire**
8. $\quad \tilde{V}_j = \mathcal{M}(V_j) - \beta_j V_{j-1}$
9. $\quad \alpha_j = \langle V_j, \tilde{V}_j \rangle_F$
10. $\quad \tilde{V}_j = \tilde{V}_j - \alpha_j V_j$
11. $\quad \beta_{j+1} = \|\tilde{V}_j\|_F$
12. $\quad V_{j+1} = \tilde{V}_j/\beta_{j+1}$
13. **Fin pour** j

On définit la matrice $\mathcal{V}_k = [V_1, ..., V_k] \in \mathbb{R}^{n \times pk}$ et $\tilde{T}_k$, la matrice tridiagonale de taille $(k+1) \times k$ dont les entrées non nulles sont calculées par l'algorithme de Lanczos symétrique global.

$$\tilde{T}_k = \begin{bmatrix} \alpha_1 & \beta_1 & 0 & \dots & 0 \\ \beta_1 & \alpha_2 & \beta_2 & \ddots & \vdots \\ 0 & \beta_2 & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \beta_{k-1} \\ \vdots & & \ddots & \ddots & \alpha_k \\ & \dots & \dots & 0 & \beta_k \end{bmatrix} \tag{4.20}$$

La matrice $T_k$ de taille $k \times k$ est obtenue en supprimant la dernière ligne de $\tilde{T}_k$. Par construction, la matrice par blocs $\mathcal{V}_k$ est F-orthonormale, ce qui signifie que les matrices $V_1, ..., V_k$ constituent un système orthonormal au sens du produit scalaire de Frobenius, c'est à dire $\langle V_i, V_j \rangle_F = \delta_{i,j},\ 1 \leq i, j \leq k$. On établit les identités suivantes de manière immédiate

**Proposition 4.5.1.** *[19] Après k itérations de l'algorithme Lanczos symétrique global, nous avons les relations suivantes :*

$$[\mathcal{M}(V_1), ..., \mathcal{M}(V_k)] = \mathcal{V}_k(T_k \otimes I_p) + E_{k+1}, \tag{4.21}$$

*où*

$$E_{k+1} = h_{k+1,k}[0_{n\times p}, ..., 0_{n\times p}, V_{k+1}],$$

*et*

$$[\mathcal{M}(V_1), ..., \mathcal{M}(V_k)] = \mathcal{V}_{k+1}(\tilde{T}_k \otimes I_p). \tag{4.22}$$

La méthode MINRES globale consiste à construire itérativement une suite $(Z_k)_{k\in\mathbb{N}^*}$ d'approximations de la solution $Z^*$ de l'équation (4.19) de la façon suivante

On commence par choisir un premier terme $X_0 \in \mathbb{R}^{n\times n}$ et en notant $\mathcal{R}_0$ le résidu correspondant : $\mathcal{R}_0 = \mathcal{C} - \mathcal{M}(Z_0)$, on construit les itérés $Z_k$ de telle manière à avoir

$$Z_k = Z_0 + \tilde{Z}_k \text{ où } \tilde{Z}_k \in \mathcal{K}_k(\mathcal{M}, \mathcal{R}_0) \tag{4.23}$$

$$\mathcal{R}_k = \mathcal{C} - \mathcal{M}(Z_k) \perp_F \mathcal{K}_k(\mathcal{M}, \mathcal{M}(\mathcal{R}_0)). \tag{4.24}$$

Par construction, le résidu $\mathcal{R}_k = \mathcal{C} - \mathcal{M}(Z_k)$ est le projeté $F$-orthogonal de $\mathcal{R}_0$ sur le sous-espace $\mathcal{K}_k(\mathcal{M}, \mathcal{M}(\mathcal{R}_0)) = \text{sev}\{\mathcal{M}(\mathcal{R}_0), ..., \mathcal{M}^k(\mathcal{R}_0)\}$ engendré par les matrices $\mathcal{M}(\mathcal{R}_0), ..., \mathcal{M}^k(\mathcal{R}_0)$. Par conséquent, la matrice $x_k$ est solution du problème de minimisation

$$\min_{Z\in Z_0+\mathcal{K}_K(\mathcal{M},\mathcal{R}_0)} \|\mathcal{C} - \mathcal{M}(Z)\|_F \tag{4.25}$$

Le résultat suivant [19] établit que le problème de moindres carrés (4.25) est équivalent à un problème de dimension réduite

**Proposition 4.5.2.** *L'approximation $Z_k$ construite par la méthode MINRES globale est donnée par*

$$Z_k = Z_0 + \mathcal{V}_k(z_k \otimes I_n),$$

*où $z_k$ est solution du problème aux moindres carrés*

$$\underset{z\in\mathbb{R}^k}{min} \| \|\mathcal{R}_0\|_F e_1 - \tilde{T}_k z\|_2 \tag{4.26}$$

*où $e_1$ est le premier vecteur de la base canonique de $\mathbb{R}^{k+1}$.*

Pour résoudre le problème (4.26), nous considérons la décomposition $QR$ de la matrice $\tilde{T}_k$

$$\tilde{T}_k \ : \ \tilde{R}_k = Q_k \tilde{T}_k,$$

où $\tilde{R}_k \in \mathbb{R}^{(k+1)\times k}$ est triangulaire supérieure et $Q_k \in \mathbb{R}^{(k+1)\times(k+1)}$ est orthogonale. Posons $g_k = \|\mathcal{R}_0\|_F Q_k e_1$. En notant $R_1$ la matrice $k \times k$ obtenue en éliminant la dernière ligne de $\tilde{R}_k$, le vecteur $y_k$ est calculé en résolvant le système triangulaire $R_1 y_k = g_k$.

A chaque itération, le résidu $\mathcal{R}_k$ doit être calculé, ce qui peut s'avérer coûteux. La proposition suivante permet de calculer $\|\mathcal{R}_k\|_F$ sans évaluer $\mathcal{M}(Z_k)$.

**Proposition 4.5.3** ([19]). *À l'étape $K$, le résidu $\mathcal{R}_k = \mathcal{C} - \mathcal{M}(Z_k)$ obtenu par la méthode MINRES globale vérifie les deux identités suivantes*

$$\mathcal{R}_k = \gamma_{k+1}\mathcal{V}_{k+1}(Q^T e_{k+1} \otimes I_p) \tag{4.27}$$

*et*

$$\|\mathcal{R}_k\|_F = |\gamma_{k+1}|, \tag{4.28}$$

*où $\gamma_{k+1}$ est la dernière composante du vecteur $g_k = \|\mathcal{R}_0\|_F Q_k e_1$ et $e_{k+1}$ est le dernier vecteur de la base canonique de $\mathbb{R}^{k+1}$ : $e_{k+1} = (0, \ldots, 0, 1)^T$.*

Dans l'algorithme MINRES globale ci-dessous. En utilisant la technique de redémarrage après un nombre choisi d'itérations. A chaque redémarrage, on choisit la dernière approximation calculée comme terme initial pour l'algorithme de MINRES global.

On peut maintenant écrire l'algorithme de la méthode MINRES global avec redémarrage :

**Algorithm 6** Algorithme MINRES global avec redémarrage

1. Initialisation : on choisit $Z_0$, une tolérance $\epsilon$ et on pose $iter = 0$
2. Calculer $\mathcal{R}_0 = \mathcal{C} - \mathcal{M}(Z_0)$, $\beta = \|\mathcal{R}_0\|$, et $V_1 = \mathcal{R}_0/\beta$
3. Calculer $\mathcal{V}_k$ et $\widetilde{T}_k$ par l'algorithme de Lanczos global appliqué à $(\mathcal{M}, V_1)$.
4. Calculer $z_k$ réalisant $\min_{z\in\mathbb{R}^k} \|\|\mathcal{R}_0\|_F e_1 - \tilde{T}_k z\|_2$
5. Calculer $Z_k = Z_0 + \mathcal{V}_k(z_k \otimes I_p)$
6. Calculer la norme du résidu $\|\mathcal{R}_k\|_F$ en utilisant la proposition (4.28)
7. **Si** $\|\mathcal{R}_k\|_F < \epsilon$
8. **Arrêt**
9. **Sinon**
10. $Z_0 = Z_k$, $\mathcal{R}_0 = \mathcal{R}_k$, $\beta = \|\mathcal{R}_0\|_F$, $V_1 = \mathcal{R}_0/\beta$, iter=iter+1, **Aller à 2** :
11. **Fin si**

la généralisation au cas $\mathcal{M}$ non symétrique est la méthode GMRES global ([19]).

## 4.6 Forme factorisée de la solution approchée

Dans cette section, on va approximer la solution approchée $X_m$ par le produit de deux matrices de rang inférieurs (low-rank approximation), pour économiser la mémoire dans les test numériques de grande taille.

À travers les exemples numériques on peut montrer que la solution exacte de l'équation de Riccati continue est de petite rang par une approximation, comme dans l'exemple suivant. Dans cet exemple, on considère le système dynamique

$$\begin{cases} \dot{\mathbf{x}}(t) &= A\,\mathbf{x}(t) + B\,u(t), \\ y(t) &= C\,\mathbf{x}(t), \end{cases}$$

où la matrice $A$ résulte de la discrétisation par différences finies de l'équation aux dérivées partielles de l'opérateur

$$L(u) = \Delta u - e^{xy}\,\frac{\partial u}{\partial x} - \sin(xy)\,\frac{\partial u}{\partial y} - (y^2 - x^2)\,u,$$

dans le domaine $\Omega = [0\ 1] \times [0\ 1]$ avec des conditions aux bord de Dirichlet homogènes.

La taille de la matrice $A$ est donnée par $n = n_0^2$, où $n_0$ est le nombre de points de la grille dans chaque direction. Pour ce test, nous prenons $n_0 = 30$ et $r = 4$. Les coefficients des matrices $B \in \mathbb{R}^{n\times r}$ et $C \in \mathbb{R}^{r\times r}$ sont des valeurs aléatoires uniformément distribuées dans l'intervalle $[0, 1]$.

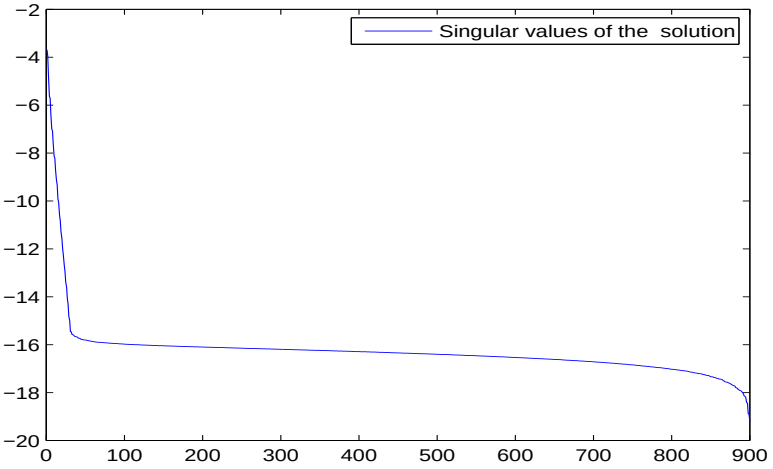


FIGURE 4.1: Valeurs singulières de la solution exacte.

La figure 4.1 représente les valeurs singulières de la solution exacte dans le cas où $n = 900$ et $r = 4$. On constate que les valeurs singulières décroissent rapidement vers zéro. Cela implique que le rang numérique de la solution exacte est petit. Dans ce cas particulier,

on a $rang(X) \simeq 25$. Donc la solution approchée $X_m$ peut être écrite comme produit de deux matrices de petit rang.

Nous considérons la décomposition en valeurs singuliers de la matrice $Y_m \in \mathbb{R}^{2mr \times 2mr}$.

$$Y_m = U \Sigma U^T$$

où $\Sigma$ est la matrice diagonale des valeurs singulières de $Y_m$ rangées dans l'ordre décroissant, $U$ est une matrice unitaire. Nous fixons une certaine tolérance *dtol* et définissons $U_l$ la matrice constituée des $l$ premières colonnes de $U$ correspondant aux $l$ valeurs singulières supérieures ou égales à *dtol*. Nous obtenons la décomposition en valeur singulière tronquée

$$Y_m \approx U_l \, \Sigma_l \, U_l{}^T$$

où $\Sigma_l = \mathrm{diag}[\sigma_1, \ldots, \sigma_l]$. Soit $Z_m = \mathbb{V}_m \, U_l \, \Sigma_l^{1/2}$, nous obtenons alors l'approximation de $X_m$ sous la forme d'une expression factorisée

$$X_m \approx Z_m \, Z_m^T. \tag{4.29}$$

Cette factorisation est très importante pour les problèmes de grande dimension, quand on n'a pas besoin de calculer l'approximation de la solution $X_m$, mais on a besoin de la stocker à chaque itération.

## 4.7 Algorithme GA-CAREs

Dans ce paragraphe, nous rappelons l'algorithme d'Arnoldi étendu par blocs en utilisant l'approche de Galerkin (GA-CARE) pour résoudre l'équation de Riccati continue de grande taille, donc l'algorithme GA-CARE est comme suit :

## 4.8 Algorithme MR-CAREs

la méthode de minimisation de résidu (MR-CAREs) pour la résolution de l'équation de Riccati continue de grande taille est résumé dans l'algorithme suivant

**Algorithm 7** Algorithme GA-CAREs

1. On choisit une tolérance $\epsilon > 0$, un nombre maximum d'itérations $m_{max}$ et une tolérance $tol_{trunc}$.
2. **Pour** $m = 1, 2, 3, ..., m_{max}$ **faire**
3. Appliquer l'algorithme 1 à $(A^T, C^T)$ pour construire les matrices $\mathbb{V}_m$ et $\mathbb{T}_m$.
4. Résoudre l'équation de Riccati

$$\mathbb{T}_m Y_m + Y_m \mathbb{T}_m^T - Y_m \tilde{B}_m \tilde{B}_m^T Y_m + \tilde{C}_m^T \tilde{C}_m = 0;$$

où $\tilde{B}_m = \mathbb{V}_m B$ et $\tilde{C}_m = \mathbb{V}_m C$.

5. Calculer la norme de Frobenius du résidu

$$r_m = \sqrt{2} \| T_{m+1,m} \tilde{Y}_m \|_F,$$

où $\tilde{Y}_m$ est la matrice de taille $2s \times 2ms$ correspondant à les $2s$ dernières lignes de la matrice de $Y_m^{GA}$.

6. **Si** $r_m < \epsilon$, **Aller à 10**.
7. **sinon** $m = m + 1$.
8. **Fin Si**
9. **Fin Pour**
10. Calculer la décomposition en valeurs et vecteurs singuliers SVD de $Y_m$, c'est-à-dire $Y_m = U \Sigma U^T$, où $\Sigma = diag[\sigma_1, \sigma_2, \cdots, \sigma_{2ms}]$ et $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_{2ms}$.
Déterminer $l$ telle que $\sigma_l \geq d_{tol} > \sigma_{l+1}$, soit $\Sigma_l = diag[\sigma_1, \sigma_2, \cdots, \sigma_l]$.
Calculer $Z_m = \mathbb{V}_m U_l \Sigma_l^{1/2}$
11. L'approximation de $X_m$ donnée par $X_m \approx Z_m Z_m^T$.

## 4.9 Exemples numériques

Dans cette section, nous présentons quelques exemples numériques de l'équation de Riccati continue de grande taille. Les algorithmes ont été codés en Matlab R2009a. Les équations de Riccati de taille réduite (4.7) ont été résolues en utilisant la fonction "care" de Matlab. Le nombre d'itérations de l'algorithme de Newton a été limité à $ite_{rmax} = 40$ et le nombre d'itérations de l'algorithme MINRES globale a été limité à $max = 1000$ avec le nombre d'itérations de redémarrage fixé à $k_{redém} = 12$. La résolution d'équations de type Lyapunov généralisée ont été effectuées jusqu'à ce que la norme du résidu soit inférieure à $tol_{MINRES} = 10^{-7}$. Le critère d'arrêt pour l'algorithme de Newton est défini par le paramètre

$$\Delta_m = \| Y_{m+1} - Y_m \| / \| Y_m \| < tol_{Newton} = 10^{-9}.$$

Les résultats suivants donnent une comparaison en résidu entre notre méthode MR-CAREs et la méthode GA-CARES [55]. Les courbes dans la suite donnent la norme du

**Algorithm 8** Algorithme MR-CAREs

1. On choisit une tolérance $\epsilon > 0$, un nombre maximum d'itérations $m_{max}$ et une tolérance $tol_{trunc}$.
2. **Pour** $m = 1, 2, 3, ..., m_{max}$ **faire**
3. Appliquer l'algorithme 1 à $(A^T, C^T)$ pour construire les matrices $\mathbb{V}_m$ et $\overline{\mathbb{T}}_m$.
4. Résoudre l'équation (4.16)
$$\Psi(Y_m^{MR}) = 0$$
5. Calculer la norme de Frobenius du résidu $r_m = \|F(Y_m^{MR})\|_F$.
6. **Si** $r_m < \epsilon$, **Aller à 10**.
7. **sinon** $m = m + 1$.
8. **Fin Si**
9. **Fin Pour**
10. Calculer la décomposition en valeurs et vecteurs singuliers SVD de $Y_m$, c'est-à-dire $Y_m = U\Sigma U^T$, où $\Sigma = diag[\sigma_1, \sigma_2, \cdots, \sigma_{2ms}]$ et $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_{2ms}$.
Déterminer $l$ telle que $\sigma_l \geq d_{tol} > \sigma_{l+1}$, soit $\Sigma_l = diag[\sigma_1, \sigma_2, \cdots, \sigma_l]$.
Calculer $Z_m = \mathbb{V}_m U_l \Sigma_l^{1/2}$
11. L'approximation de $X_m$ donnée par $X_m \approx Z_m Z_m^T$.

résidu $\| R_m \|_F$, où $R_m = A^T X + XA - XBB^T + C^T C$ dans une échelle logarithmique à chaque itération $m$.

### 4.9.1 Exemple 1

Dans tous les tests effectués dans sous section, la matrice $A$ provenant de la discrétisation de l'opérateur

$$L_u = \Delta u - f_1(x,y)\frac{\partial u}{\partial x} + f_2(x,y)\frac{\partial u}{\partial y} + g(x,y), \tag{4.30}$$

dans le domaine $[0,1] \times [0,1]$ avec les condition de Dirichlet homogène. La taille de la matrice construite est $n = n_0^2$, où $n_0$ est le nombre de points de la grille dans chaque direction. La discrétisation de l'opérateur $L_u$ construites des matrices dans la bibliothèque `Lyapack` [89] et désignés par

A = fdm2d_matrix(n0,'f_1(x,y)','f_2(x,y)','g(x,y)').

dans le domaine $\Omega = [0\ 1] \times [0\ 1]$ avec des conditions aux bord de Dirichlet homogènes, où les fonctions $f\_1$, $f\_2$ et $g$ sera précisées dans chaque exemple. Pour tous les test, nous prenons $r = 2$.

#### Exemple 1.1

Dans cet exemple, nous considérons $n = 25000$ et

A= fdm2d_matrix(50,'$x+y$','$exp(x*y)$','10')

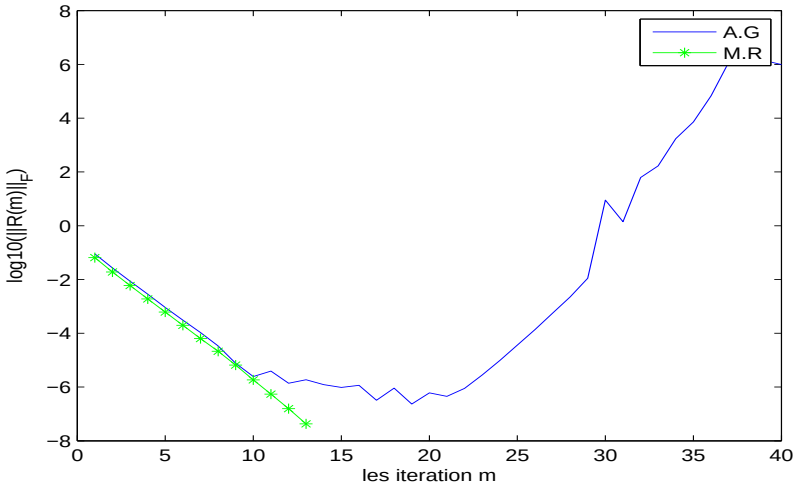


FIGURE 4.2: Résultats de l'exemple 1.1

**Exemple 1.2**

Dans cet exemple, nous considérons $n = 64000$ et la matrice $A$

A= fdm2d_matrix(80,'$cos(x+y)$','$sin(y^2)$','$y^2-x^2$')

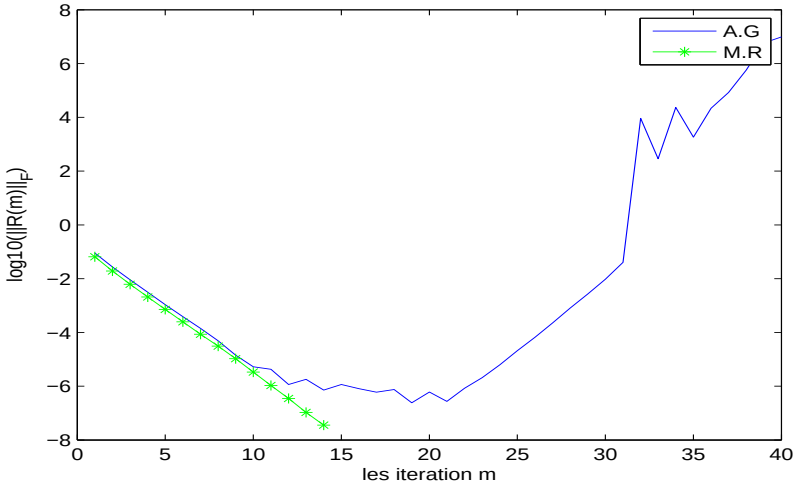


FIGURE 4.3: Résultats de l'exemple 1.2.

### 4.9.2 Exemple 2

Nous considérons l'exemple donné par Jbilou [64, 65] tel que la matrice $A$ est donnée par,

$$A = -\begin{pmatrix} a & 1-d & \cdots & 0 & 1 \\ 1+d & a & \ddots & \cdots & 0 \\ 0 & 1+d & \ddots & \ddots & \vdots \\ 0 & & \ddots & \ddots & 1-d \\ 1 & 0 & 0 & 1+d & a \end{pmatrix}.$$

Pour $n = 4000$, $C = I_{n\times s}$, $B = rand(n, 10)$, et $d = 0.2$, nous obtenons le résultat suivant,

| Méthode | Itér | Temps | $\| R_m \|_F$ |
|---|---|---|---|
| MR | 5 | 13 s | $9.00 \times 10^{-9}$ |
| GA | 24 | 24 s | $4.17 \times 10^{-4}$ |

TABLE 4.1: Résultats de l'exemple 2

Notons que l'algorithme GA s'arrête à l'itération $m = 25$ : c'est un problème d'existence et d'unicité de la solution de l'équation de Riccati projetée.

### 4.9.3 Exemple 3

Soit $A$ la matrice opposée de la matrice précédente. Pour $d = 0, 5$, nous avons

| Méthode | N.Itérations | Temps | $\| R_m \|_F$ |
|---|---|---|---|
| MR | 7 | 11 s | $2.17 \times 10^{-10}$ |
| GA | 11 | 6 s | $1.39 \times 10^{3}$ |

TABLE 4.2: Résultats de l'exemple 3.

Notons que l'algorithme GA s'arrête à l'itération $m = 12$ : c'est un problème d'existence et d'unicité de la solution de l'équation de Riccati projetée.

## 4.10 Conclusion

Dans ce chapitre, nous avons proposé une nouvelle méthode pour la résolution numérique de l'équation de Riccati continue de grande taille. Basée sur la méthode de minimisation de résidu MR (Minimal Residual) et la méthode de projection dans un sous-espaces de Krylov étendu par blocs, elle se ramène à chaque itérations à la résolution numérique d'un problème de minimisation réduit. Ce dernier problème est résolu à l'aide de la

méthode de Newton et la méthode de MINRES globale. Les tests numériques montrent l'efficacité de notre méthode.

Chapitre 5

# Équation matricielle de Riccati non symétrique et application à l'équation de transport

## 5.1 Introduction

Les équations algébriques de Riccati non symétriques NAREs (Nonsymmetric Algebraic Riccati Equation) jouent un rôle fondamental dans de nombreux problèmes de la théorie des jeux différentiels, des modèles de Markov, et dans la théorie de transport [1, 24, 26, 32, 38, 41, 46, 71, 72, 110]. L'équation de Riccati non symétrique est de la forme suivante

$$\mathcal{R}(X) = XCX - XD - AX + B = 0 \ \text{(NARE)} \tag{5.1}$$

où $A \in \mathbb{R}^{n\times n}$, $D \in \mathbb{R}^{p\times p}$, $B \in \mathbb{R}^{n\times p}$, et $C \in \mathbb{R}^{p\times n}$. On peut associer l'équation de Riccati non symétrique (5.1) à la matrice $\mathcal{M}$ qui se présente sous la forme suivante

$$\mathcal{M} = \begin{pmatrix} D & -C \\ -B & A \end{pmatrix} \in \mathbb{R}^{(n+p)\times(n+p)}. \tag{5.2}$$

On suppose que $\mathcal{M}$ est une M-matrice inversible. Nous rappelons que $\mathcal{M}$ est une M-matrice si elle peut être écrite comme $\mathcal{M} = aI - \mathcal{N}$ où $\mathcal{N}$ matrice avec des éléments non-négatifs et $a \geq \rho(\mathcal{N})$, où $\rho(\mathcal{N})$ est le rayon spectral de la matrice $\mathcal{N}$. Cela assure l'existence de la solution non négative minimale $X^*$ de l'équation de Riccati non symétrique (5.1), pour plus de détails voir [24, 25, 41, 46, 71, 72].
Soit $\mathcal{H}$ la matrice définie par

$$\mathcal{H} = J\mathcal{M}$$

où $J = \begin{pmatrix} I_n & 0 \\ 0 & -I_p \end{pmatrix}$, avec $I_n$ la matrice identité de taille $n$. Donc nous avons

$$\mathcal{H} = \begin{pmatrix} D & -C \\ B & -A \end{pmatrix}. \tag{5.3}$$

La solution de l'équation (5.1) est liée au calcul de sous espaces invariants de la matrice $\mathcal{H}$ associée. En effet, si $X$ est une solution de l'équation de Riccati non symétrique (5.1), alors on a

$$\mathcal{H}\begin{pmatrix} I_n \\ X \end{pmatrix} = \begin{pmatrix} I_n \\ X \end{pmatrix}(A - XB)$$

Si les matrices $Y$, $Z$ et $W$ dans $\mathbb{R}^{n\times n}$, avec $Z$ non singulière, satisfont :

$$\mathcal{H}\begin{pmatrix} Y \\ Z \end{pmatrix} = \begin{pmatrix} Y \\ Z \end{pmatrix} W,$$

alors $X = YZ^{-1}$ est une solution de l'équation algébrique de Riccati non symétrique, pour plus de détails sur les propriétés de la matrice $\mathcal{H}$ et l'équation de Riccati non symétrique voir [24–26, 41, 46, 110].

Le théorème de l'existence et l'unicité des solutions de Riccati non symétrique est liée à la matrice $\mathcal{M}$ voir [24, 41, 46, 71, 72]. De plus, la structure particulière de la matrice $\mathcal{M}$ assure l'existence de la solution minimale non négative $X^*$ tel que $X^* \geq 0$ et $X \geq X^*$ pour toute solution $X$ de l'équation de Riccati non symétrique (5.1), pour plus de détails voir [24, 41, 46]. Cela est indiqué dans le théorème suivant

**Théorème 5.1.1.** *[46] Supposons que $\mathcal{M}$ soit une M-matrice. Alors l'équation NARE (5.1) admet une unique solution minimale non négative $X^*$. De plus, si $\mathcal{M}$ est inversible, alors $A - X^*C$ et $D - CX^*$ sont inversibles et M-matrices.*

Pour les équations NAREs de petites ou moyennes tailles, il y a plusieurs méthodes numériques pour trouver la solution non-négative minimale $X^*$ de l'équation de Riccati non symétrique. Par exemple, les méthodes exactes de Schur ont été étudiées dans [41, 48, 49]. La méthode de Newton a également été étudiée dans [26, 41, 46, 48], elle nécessite à chaque étape de calculer la solution d'une équation matricielle de Sylvester. La méthode peut être coûteuse lorsqu'on résout l'équation de Sylvester par les méthodes directs. D'autres méthodes comme l'algorithme SDA (Structure-preserving Doubling Algorithm) [50] et la méthode ALI (Alternately Linearized Implicit) [10] ont été proposées ces dernières années. En général, les méthodes itératives du point-fixe [25, 41, 46, 84–86] sont moins coûteuses que la méthode de Newton, la méthode de Schur ou les méthodes

SDA. Certaines techniques d'accélération de la convergence basées sur des méthodes d'extrapolation vectorielles [69] ont été proposées récemment dans [38] pour accélérer la convergence de certaines de ces méthodes itératives de point fixe, comme celles introduites dans [38, 40, 47].

Dans ce chapitre, nous considérons l'équation matricielle de Riccati non symétrique (NAREs) de grande taille avec le second membre s'écrit comme le produit de deux matrices de petit rang. Ce type d'équations NAREs intervient notamment dans la théorie du transport et ses applications. Nous nous proposons dans ce chapitre, deux nouvelles méthodes pour la résolution numérique de l'équation de Riccati non symétrique (NAREs) de grande taille. La première est une méthode itérative de projection sur des sous espaces de Krylov étendu EBA (Extended Block Arnoldi) [36, 56, 102]. Pour obtenir des solutions approchées de rang inférieur "Low-rank", nous allons traiter aussi le cas particulier correspondant à l'équation matricielle de Riccati non symétrique NAREs dans la théorie de transport [38, 72].

La deuxième méthode, consiste à combiner la méthode de Newton-Kleinman et la méthode d'Arnoldi par blocs. A chaque étape la méthode de Newton-Kleinman nécessitera la résolution d'une équation matricielle de Sylvester de grande taille. Cette équation ne pouvait pas être résolu par les méthodes directes comme l'algorithme Bartels-Stewart [11]. On peut utiliser les méthodes de de type Krylov, comme celles définies dans [29, 56, 62, 65], pour résoudre l'équation de Sylvester de grande taille.

Le reste de ce chapitre est organisé comme suit : Dans la Section 2, nous allons combiner la méthode de Newton-Kleinman avec les méthodes de sous-espaces de Krylov par bloc ou bien des sous-espaces de Krylov étendus par blocs. La Section 3 est consacrée à la résolution de l'équation NAREs de grande taille par une méthode de projection à l'aide du sous espace de Krylov étendu et la condition d'orthogonalité de de Galerkin. Les solutions approchées de l'équation NAREs sont données sous la forme factorisée (low rank approximation). L'application aux équations algébriques de Riccati non symétriques dans la théorie de transport sera considérée dans la Section 4. Enfin, la dernière partie est consacrée à quelques expériences numériques.

Dans ce chapitre, nous utilisons les notations suivantes : Si $K$ et $L$ sont deux matrices de taille $n \times m$, alors $K \geq L$ si $L_{i,j} \geq K_{i,j}$ pour tout $i$, $j$. La matrice $K$ est dite non négative si $L_{i,j} \geq 0$. Enfin, la séparation, par rapport à la norme de Frobenius, entre deux matrices $K$ et $L$ est donnée par

$$\mathrm{sep}_F(K, L) = \min_{\|X\|_F=1} \|KX - XL\|_F.$$

## 5.2 La méthode de Newton-Krylov par blocs

La méthode de Newton-Kleinman a été introduite par D.L. Kleinman en 1968 [79], pour résoudre l'équation de Riccati continue, basée sur une linéarisation préliminaire par la méthode de Newton, qui nous ramène à chaque étape à la résolution d'une équation matricielle de Lyapunov avec un second membre décomposé (low rank). Nous nous attachons dans ce paragraphe à résoudre l'équation de Riccati non symétrique. La méthode de Newton nécessite à chaque itération la résolution d'une grande équation de Sylvester. Cette équation est résolue par la méthode de Galerkin basée sur l'algorithme d'Arnoldi par blocs. On suppose dans cette partie que

$$B, C \neq 0,\ (I \otimes A + D^T \otimes I)Vect(B) > 0 \tag{5.4}$$

et $(I \otimes A + D^T \otimes I)$ inversible et $M$-matrice. Ces conditions sont suffisantes pour assurer la convergence de la méthode de Newton avec l'itération initiale de Newton $X_0 = 0$.

Soit

$$\mathcal{R}(X) = -XC_1C_2^TX + XD + AX - EF^T. \tag{5.5}$$

Nous appliquons la méthode de Newton-Kleinman pour résoudre l'équation

$$\mathcal{R}(X) = 0.$$

La dérivée de Fréchet de l'application non linéaire $\mathcal{R}(X)$ en $X_k$ est donnée par

$$\mathcal{R}'_{X_k}(Z) = (A - X_kC_1C_2^T)Z + Z(D - C_1C_2^TX_k). \tag{5.6}$$

En choisissant comme premier terme $X_0 = 0_{n\times n}$, on construit la suite $(X_k)_{k\in\mathbb{N}}$ des approximations de la solution de (5.5) définie par la relation de récurrence

$$\mathcal{R}'_{X_k}(X_{k+1} - X_k) = -\mathcal{R}(X_k). \tag{5.7}$$

On montre facilement que les itérations $X_{k+1}$ de Newton peuvent être vues comme solutions de l'équation matricielle de Sylvester

$$A_k\, X_{k+1} + X_{k+1}\, D_k + L_kM_k^T = 0 \tag{5.8}$$

où $A_k = A - X_kC_1C_2^T$, $D_k = D - C_1C_2^TX_k$, $L_k = [X_kC_1\ ,\ -E]$ et $M_k = [X_k^TC_2\ ,\ F]$.

La convergence de la méthode de Newton pour la résolution de l'équation de Riccati non symétrique est donnée par le théorème suivant ;

**Théorème 5.2.1** (Guo et Higham,(2007) [46]). *Si $X^*$ la solution non négative minimale de l'équation (5.14), alors les itérés de la méthode de Newton $\{X_i\}$ sont bien défini, avec*

$X_0 = 0$, *et nous avons*

$$\forall k \geq 1,\ X_0 \leq X_1 \leq X_k \leq X^*,\ \text{ et } \lim_{k\to\infty}(X_k) = X^*.$$

*De plus, si la matrice $\mathcal{M}$ est inversible la convergence est quadratique.*

L'algorithme suivant décrit le processus de la méthode de Newton-Kleinman pour résoudre l'équation de Riccati non symétrique

**Algorithm 9** [La Méthode de Newton-Kleinman pour résoudre NAREs]

– On choisit une estimation initiale $X_0$, une tolérance *tol* et on fixe *kmax*.
– Pour $k = 0, \ldots, kmax$
  – Calculer $X_{k+1}$ la solution de l'équation de Sylvester de grande taille suivante

$$A_k\, X_{k+1} + X_{k+1}\, D_k + L_k M_k^T = 0 \tag{5.9}$$

  – Si $\| \mathcal{R}(X_{k+1}) \|_F <$ *tol*, Arrêt
– Fin pour.

Dans chaque itération de l'algorithme de Newton-Kleinman, on résout l'équation matricielle de Sylvester avec le second membre donné comme le produit de deux matrices de petit rang. Pour les petites et moyennes dimensions, on peut utiliser des méthodes directes comme la méthode Bartels-Stewart [11]. Dans le cas où $n$ est grand, les méthodes de projection sur des espaces de type Krylov, plusieurs méthodes numériques ont été proposées ces dernières années ; voir [29, 55, 56, 64, 65, 100, 102]. Ici, nous avons utilisé des sous espaces de Krylov étendus par blocs (ou le sous espaces de Krylov par bloc) pour résoudre l'équation matricielle de Sylvester de grande taille (5.8). Le procédé est défini comme suit : On applique d'abord l'algorithme d'Arnoldi étendu par blocs (ou l'algorithme d'Arnoldi par blocs) à $(A_k, L_k)$ et $(D_k^T, M_k)$ à l'étape $m$, il permet de générer, les matrices orthonormés $\mathrm{V}_{m,k}$ et $\mathrm{W}_{m,k}$ dont les colonnes sont des bases orthonormales du sous espace de Krylov étendu par blocs de $\mathcal{K}_m(A_k, L_k)$ et $\mathcal{K}_m(D_k^T, M_k)$, respectivement. Les approximations $X_{k+1}^m$ de la solution $X_{k+1}$ sont données par :

$$X_{k+1}^m = \mathrm{V}_{m,k} Z_m \mathrm{W}_{m,k}^T, \tag{5.10}$$

et $Z_m$ est obtenu à partir de la condition d'orthogonalité de Galerkin :

$$\mathrm{V}_{m,k}^T\, \mathcal{R}(X_{k+1}^m)\, \mathrm{W}_{m,k} = 0, \tag{5.11}$$

où $\mathcal{R}(X_{k+1}^m) = A_k\, X_{k+1}^m + X_{k+1}^m\, D_k + L_k M_k^T$ est le résidu de $X_{k+1}^m$ correspondant à l'équation de Sylvester (5.8). Par conséquent, nous obtenons la matrice $Z_m$ est la solution de l'équation de Sylvester de petit taille suivante :

$$\mathrm{T}_m^{(A)} Z_m + Z_m {\mathrm{T}_m^{(D)}}^T + \mathrm{L}_m \mathrm{M}_m^T = 0, \tag{5.12}$$

où $\mathrm{T}_m^{(A)} = \mathrm{V}_{m,k}^T A_k \mathrm{V}_{m,k}$, $\mathrm{T}_m^{(D)} = \mathrm{W}_{m,k}^T D_k^T \mathrm{W}_{m,k}$, $\mathrm{L}_m = \mathrm{V}_{m,k}^T L_k$ et $\mathrm{M}_m = \mathrm{W}_{m,k}^T M_k$. L'équation de Sylvester de taille réduite (5.12) sera résolu par la méthode de Bartels-Stewart [11]. On peut calculer la norme du résidu $\| \mathcal{R}(X_{k+1}^m) \|_F$ sans avoir calculer la solution approchée $X_{k+1}^m$, voir [56, 62] pour plus de détails. Ici l'approximation $X_{k+1}^m$ peut être exprimée comme un produit de deux matrices de rang inférieur et cela permet d'économiser de la place mémoire.

Notons que dans le cas de l'utilisation de l'algorithme d'Arnoldi étendu par blocs pour résoudre l'équation matricielle de Sylvester de grande taille (5.9), nous avons besoin de calculer produits de la forme $A_k^{-1}Y$ et $D_k^{-T}Y$ où $A_k = A - X_k C_1 C_2^T$ et $D_k = D - C_1 C_2^T X_k$ avec $X_k$ sous la forme suivante $X_k = Z_1 Z_2^T$. Puisque $A$ et $D$ sont creuses, les matrices $A_k$ et $D_k$ sont pas plus creuses, puis le calcul de les produits $A_k^{-1}Y$ et $D_k^{-T}Y$ devient très coûteux. La meilleure méthode pour éviter ce problème est d'utiliser la formule de Sherman-Morrison-Woodbury donnée par

$$(A + UV^T)^{-1} Y = A^{-1}Y - A^{-1}U(I + V^T A^{-1}U)V^T A^{-1}Y, \tag{5.13}$$

où $U$ et $V$ sont des matrices de taille $n \times r$. Notons que, si nous utilisons la méthode d'Arnodi par bloc [29, 62] pour résoudre l'équation matricielle de Sylvester (5.9), alors on a seulement les produits de la forme $A_k Y$ et $D_k^T Y$ qui sont nécessaires. Généralement, cette méthode nécessite plusieurs itérations, par rapport à la méthode d'Arnodi étendu par bloc, pour obtenir de bonnes approximations mais pourrait être moins cher si les produits inverse "matrice-matrice" sont dominants.

## 5.3 Solution approchée de rang inférieur de NAREs

Dans cette section, nous considérons les équations algébriques de Riccati non symétriques avec un second membre donné comme produit de deux matrices $E$ et $F$ de petit rang. Dans ce cas l'équation de Riccati non symétrique est de la forme suivante :

$$AX + XD - XC_1C_2^T X - EF^T = 0, \quad \text{(LrNARE)} \tag{5.14}$$

où les matrices $A$ et $D$ sont de grande taille et inversibles. $E \in \mathbb{R}^{n\times s}$, $F \in \mathbb{R}^{p\times s}$, $C_1 \in \mathbb{R}^{p\times s}$ et $C_2 \in \mathbb{R}^{n\times s}$ avec $s \ll n$, $p$. Ces équations matricielles présentent de nombreuses applications telles que dans la théorie de transport et autres. A titre d'exemple, nous considérons l'équation de Riccati non symétrique de la théorie de transport avec $n = p = 900$ et $c = \alpha = 0.5$ (voir la section suivante). Sur la figure 5.1, nous avons tracé la décomposition en valeurs singulières (SVD) de la solution non négative minimale exacte obtenue par la méthode de Schur.

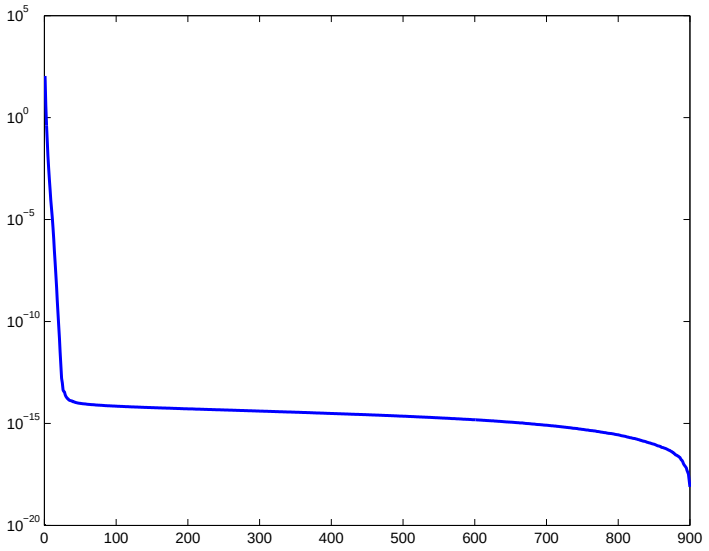


FIGURE 5.1: Les valeurs singulières de la solution non négative minimale de l'équation NAREs

Comme le montre cette figure, les valeurs singulières décroisent rapidement vers zéro, donc le rang numérique de la solution exacte (non négative minimale) est petit, ce qui donne une solution de rang inférieur "low-rank". Ici on a $rang(X^*) = 20$. Cette remarque permet de rechercher des méthodes qui donne des approximations de petit rang "low-rank" de la solution de l'équation LrNARE.

Dans la suite, nous proposons une méthode itérative de projection sur des sous espaces de Krylov étendu par blocs pour résoudre l'équation LrNARE, basée sur l'algorithme d'Arnoldi étendu par blocs et la condition d'orthogonalité de Galerkin. Cette méthode est décrite ci-dessous comme suit :

Nous appliquons l'algorithme d'Arnoldi étendu par blocs à $(A, E)$ et $(D^T, F)$, nous obtenons les matrices orthogonales $\mathbb{V}_{m+1}$ et $\mathbb{W}_{m+1}$ et les matrices de Hessenberg supérieure par blocs $\mathbb{H}_m^A$ et $\mathbb{H}_m^D$. Nous définissons aussi les matrices de Hessenberg supérieures par blocs suivantes

$$\mathbb{T}_m^A = \mathbb{V}_m^T A \mathbb{V}_m, \quad \overline{\mathbb{T}_m^A} = \mathbb{V}_{m+1}^T A \, \mathbb{V}_m,$$

et

$$\mathbb{T}_m^D = \mathbb{W}_m^T D^T \mathbb{W}_m, \quad \overline{\mathbb{T}_m^D} = \mathbb{W}_{m+1}^T D^T \, \mathbb{W}_m.$$

Comme nous avons expliqué précédemment que les matrices Hessenberg par blocs $\mathbb{T}_m^A$ et $\mathbb{T}_m^D$ sont obtenus à partir $\mathbb{H}_m^A$ et $\mathbb{H}_m^D$ sans avoir calculer le produit "matrice-vecteur". Ensuite, nous considérons des solutions approchées de petit rang de la forme

$$X_m = \mathbb{V}_m \, Y_m \, \mathbb{W}_m^T, \tag{5.15}$$

avec $\mathbb{V}_m = [V_1, \ldots, V_m]$, $\mathbb{W}_m = [W_1, \ldots, W_m]$ et $Y_m \in \mathbb{R}^{2ms \times 2ms}$. La matrice $Y_m$ est obtenue à partir de la condition de Galerkin suivante

$$\mathbb{V}_m^T \mathcal{R}(X_m) \mathbb{W}_m = 0, \tag{5.16}$$

où $\mathcal{R}(X_m)$ est le résidu correspondant à l'approximation $X_m$ et défini par :

$$\mathcal{R}(X_m) = AX_m + X_m D - X_m C_1 C_2^T X_m - EF^T.$$

Nous utilisons la condition d'orthogonalité de Galerkin (5.16) et le fait que les matrices $\mathbb{V}_m$ et $\mathbb{W}_m$ sont orthogonales, nous obtenons une équation projetée de type Riccati non symétrique

$$\mathbb{T}_m^A Y_m + Y_m (\mathbb{T}_m^D)^T - Y_m C_{1,m} C_{2,m}^T Y_m - \widetilde{E}_m \widetilde{F}_m^T = 0, \tag{5.17}$$

où $C_{1,m} = \mathbb{W}_m^T C_1$, $C_{2,m} = \mathbb{V}_m^T C_2$, $\widetilde{E}_m = \mathbb{V}_m^T E$ et $\widetilde{F}_m = \mathbb{W}_m^T F$.

Notons que si $\mathcal{M}_m$ la matrice associée a l'équation projetée (5.17), c'est à dire

$$\mathcal{M}_m = \begin{pmatrix} (\mathbb{T}_m^D)^T & -C_{1,m} C_{2,m}^T \\ -\widetilde{E}_m \widetilde{F}_m^T & \mathcal{T}_m^A \end{pmatrix}.$$

Soit la matrice orthogonale $\mathcal{Q}_m$ définie par :

$$\mathcal{Q}_m = \begin{pmatrix} \mathcal{W}_m & 0 \\ 0 & \mathcal{V}_m \end{pmatrix}.$$

Alors nous avons la relation entre la matrice $\mathcal{M}$ associée à l'équation (5.14) et la matrice $\mathcal{M}_m$ :

$$\mathcal{M}_m = \mathcal{Q}_m^T \, \mathcal{M} \, \mathcal{Q}_m. \tag{5.18}$$

Nous supposons que l'équation projetée de Riccati non symétrique (5.17) à une solution non négative minimale unique qui pourrait être obtenue par des méthodes classiques.
Nous avons le résultat suivant qui nous permet de calculer la norme du résidu sans calculer la solution approchée $X_m$.

**Théorème 5.3.1.** *Soit $Y_m$ la solution exacte de l'équation NARE réduite (5.17), obtenue à l'itération $m$ de l'algorithme d'Arnoldi étendu par blocs . Alors la norme de Frobenius du résidu $\mathcal{R}_m = \mathcal{R}(X_m)$ est donnée par :*

$$\| \mathcal{R}_m \|_F^2 = \| Y_m \mathbb{E}_m (\mathbb{T}_{m+1,m}^D)^T \|_F^2 + \| \mathbb{T}_{m+1,m}^A \mathbb{E}_m^T Y_m \|_F^2, \tag{5.19}$$

*où $\mathbb{T}_{m+1,m}^A$ et $\mathbb{T}_{m+1,m}^D$ sont les dernier bloc des matrices $\overline{\mathbb{T}_m^A}$ et $\overline{\mathbb{T}_m^D}$ respectivement.*

*Démonstration.* Soit $\mathbb{V}_{m+1} = [\mathbb{V}_m, V_{m+1}]$ et $\mathbb{W}_{m+1} = [\mathbb{W}_m, W_{m+1}]$ les matrices orthonormales construites en appliquant simultanément $m$ itérations de l'algorithme EBA aux paires de matrices $(A, E)$ et $(D^T, F)$ respectivement, donc on a $\mathbb{T}_m^A = \mathbb{V}_m^T A \mathbb{V}_m$ et $\mathbb{T}_m^D = \mathbb{W}_m^T D^T \mathbb{W}_m$. Le résidu $\mathcal{R}(X_m)$ peut être formulé comme un produit de matrices

$$\begin{aligned}\mathcal{R}(X_m) &= AX_m + X_m D - X_m C_1 C_2^T X_m - EF^T \\ &= \mathbb{V}_{m+1} \begin{bmatrix} \mathbb{T}_m^A Y_m + Y_m (\mathbb{T}_m^D)^T - Y_m \widetilde{C}_1 \widetilde{C}_2^T Y_m - \widetilde{E}\widetilde{F}^T & Y_m \mathbb{E}_m (T_{m+1,m}^D)^T \\ T_{m+1,m}^A \mathbb{E}_m^T Y_m & 0_{2r} \end{bmatrix} \mathbb{W}_{m+1}\end{aligned}$$

où $\widetilde{C}_1 = \mathbb{W}_m^T C_1$, $\widetilde{C}_2 = \mathbb{V}_m^T C_2$, $\widetilde{E} = \mathbb{V}_m^T E$ et $\widetilde{F} = \mathbb{W}_m^T F$.
Comme $Y_m$ est la solution de l'équation projetée

$$\mathbb{T}_m^A Y_m + Y_m (\mathbb{T}_m^D)^T - Y_m \widetilde{C}_1 \widetilde{C}_2^T Y_m - \widetilde{E}\widetilde{F}^T = 0,$$

nous avons

$$\mathcal{R}_m = \mathcal{R}(X_m) = \mathbb{V}_{m+1} \begin{pmatrix} 0 & Y_m \mathbb{E}_m (T_{m+1,m}^D)^T \\ T_{m+1,m}^A \mathbb{E}_m^T Y_m & 0 \end{pmatrix} \mathbb{W}_{m+1}^T. \tag{5.20}$$

Puisque $\mathbb{V}_{m+1}$ et $\mathbb{W}_{m+1}$ sont orthonormales, nous obtenons :

$$\| \mathcal{R}_m \|_F^2 = \| Y_m \mathbb{E}_m T_{m+1,m}^{BT} \|_F^2 + \| T_{m+1,m}^A \mathbb{E}_m^T Y_m \|_F^2 .$$

□

Le théorème précédent joue un rôle très important dans la pratique, il permet de calculer la norme du résidu sans calculer l'approximation de la solution et de minimiser le temps d'exécution.

pour économiser de la mémoire, la solution approchée $X_m = \mathbb{V}_m Y_m \mathbb{W}_m^T$ pourrait être donné comme un produit de deux matrices de petit rang. En effet, considérons la décomposition en valeurs singulières de la matrice $Y_m$, c'est-à-dire

$$Y_m = \widetilde{U}\, \Sigma\, \widetilde{V}^T,$$

où $\Sigma$ est la matrice diagonale des valeurs singulières de $Y_m$ rangées dans l'ordre décroissant. Soient $\widetilde{U}_l$ et $\widetilde{V}_l$ les matrices constituées des $l$ premières colonnes de $\widetilde{U}$ et de $\widetilde{V}$ correspondant aux $l$ valeurs singulières supérieures ou égales certaine tolérance *dtol*. Nous obtenons la décomposition de valeur singulière tronquée $Y_m \approx \widetilde{U}_l\, \Sigma_l\, \widetilde{V}_l^T$ où $\Sigma_l = \mathrm{diag}[\sigma_1, \ldots, \sigma_l]$. Soit $\mathcal{Z}_m^{(1)} = \mathbb{V}_m\, \widetilde{U}_l\, \Sigma_l^{1/2}$, et $\mathcal{Z}_m^{(2)} = \mathbb{W}_m\, \widetilde{V}_l\, \Sigma_l^{1/2}$, il en résulte que

$$X_m \approx \mathcal{Z}_m^{(1)} \left(\mathcal{Z}_m^{(2)}\right)^T. \tag{5.21}$$

Cette factorisation est très importante pour les problèmes de grande dimension, quand on n'a pas besoin de calculer l'approximation de la solution $X_m$, mais on a besoin de la stocker à chaque itération.

Nous donnons maintenant un résultat de perturbation

**Théorème 5.3.2.** *Soit $X_m$ l'approximation de rang inférieur de l'équation LrNARE (5.14). Nous avons :*

$$(A - K_m)X_m + X_m(D - J_m) - X_m C_1 C_2^T X_m - EF^T = 0, \tag{5.22}$$

*où $K_m = V_{m+1}T^A_{m+1,m}V_m^T$ et $J_m = W_m(T^D_{m+1,m})^T W_{m+1}^T$.*

*Démonstration.* En multipliant l'équation de Riccati non symétrique de dimension réduite (1.5) à gauche et à droite par $\mathbb{V}_m$ et $\mathbb{W}_m^T$ respectivement, et en utilisant les les relations suivante

$$A\mathbb{V}_m = \mathbb{V}_m\mathbb{T}_m^A + V_{m+1}T^A_{m+1,m}\mathbb{E}_m^T,$$

et

$$D^T\mathbb{W}_m = \mathbb{W}_m\mathbb{T}_m^D + W_{m+1}T^D_{m+1,m}\mathbb{E}_m^T.$$

nous obtenons

$$[A\mathbb{V}_m - V_{m+1}T^A_{m+1,m}\mathbb{E}_m^T]Y_m\mathbb{W}_m^T + \mathbb{V}_mY_m[D^T\mathbb{W}_m - W_{m+1}T^D_{m+1,m}\mathbb{E}_m^T]^T$$

$$-\mathbb{V}_mY_m\mathbb{W}_m^TBC^T\mathbb{V}_mY_m\mathbb{W}_m^T + EF^T = 0.$$

Comme $\mathbb{V}_m$ et $\mathbb{W}_m$ sont deux matrices orthonormées, cela nous permet de donner les relations suivantes

$$V_{m+1}T^A_{m+1,m}\mathbb{E}_m^TY_m\mathbb{W}_m^T = V_{m+1}T^A_{m+1,m}\mathbb{E}_m^T\mathbb{V}_m^TX_m$$

et

$$\mathbb{V}_mY_m\mathbb{E}_m(T^D_{m+1,m})^TW_{m+1}^T = X_m\mathbb{W}_m\mathbb{E}_m(T^D_{m+1,m})^TW_{m+1}^T$$

d'autre part, on a également

$$\mathbb{V}_m\mathbb{E}_m = V_m \text{ et } \mathbb{W}_m\mathbb{E}_m = W_m.$$

Par conséquent

$$(A - K_m)X_m + X_m(D - J_m) - X_mBC^TX_m + EF^T = 0,$$

où $K_m = V_{m+1}T^A_{m+1,m}V_m^T$ et $J_m = W_m(T^D_{m+1,m})^TW_{m+1}^T$. □

Lorsque $D = A^T$, $C_1 = C_2$ et $E = F$ l'équation (5.14) sera une équation de Riccati symétrique, et le résultat du théorème 5.3.2 coïncide dans ce cas avec le résultat de la perturbation donnée dans l'article de Jbilou [65].

Ensuite, nous donnons un résultat montrant que l'erreur $X - X_m$ est une solution exacte de l'équation NARE perturbée.

**Théorème 5.3.3.** *Soit $X_m$ la solution approchée obtenue par m itération de l'algorithme d'Arnoldi étendu par blocs, et X la solution de l'équation LrNARE (5.14). Alors, l'erreur $X - X_m$ est une solution de l'équation algébrique de Riccati non symétrique perturbée suivante*

$$A_m(X - X_m) + (X - X_m)D_m - (X - X_m)C_1C_2^T(X - X_m) + K_mX_m + X_mJ_m = 0 \quad (5.23)$$

*où $A_m = A - X_m C_1C_2^T$ et $D_m = D - C_1C_2^T X_m$, $K_m = V_{m+1}\mathbb{T}^A_{m+1,m}V_m^T$ et $J_m = W_m(\mathbb{T}^D_{m+1,m})^T W_{m+1}^T$.*

*Démonstration.* D'après le théorème (5.3.2), nous avons :

$$(A - K_m)X_m + X_m(D - J_m) - X_mC_1C_2^TX_m - EF^T = 0. \quad (5.24)$$

La soustraction (5.24) et (5.14), permet de donner l'équation suivante

$$A(X - X_m) + (X - X_m)D - XC_1C_2^TX + X_mC_1C_2^TX_m + K_mX_m + X_mJ_m = 0. \quad (5.25)$$

Enfin, en développant (5.25), nous obtenons (5.23) se qui termine la démonstration. □

Nous donnons maintenant un résultat de majoration de la norme de l'erreur $X - X_m$ au pas $m$ de l'algorithme d'Arnoldi étendu par blocs.

**Théorème 5.3.4.** *Soit $X_m$ l'approximation de faible rang de la solution exacte X de l'équation LrNARE. Si $\delta_m = \mathrm{sep_F}(A_m, -D_m) > 0$ et $\frac{\gamma_m\eta}{\delta_m^2} < 1/4$, il existe une solution X de (5.14) satisfaisant*

$$\| X - X_m \|_F \leq \frac{2\gamma_m}{\delta_m + \sqrt{\delta_m^2 - 4\gamma_m\eta}}.$$

*où $\gamma_m = \| R_m \|_F$ et $\eta = \| C_1C_2^T \|_F$.*

*Démonstration.* Tout d'abord, remarquer que

$$K_mX_m + X_mJ_m = \mathbb{V}_{m+1}\begin{pmatrix} 0 & Y_m\mathbb{E}_m(\mathbb{T}^D_{m+1,m})^T \\ \mathbb{T}^A_{m+1,m}\mathbb{E}_m^TY_m & 0 \end{pmatrix}\mathbb{W}_{m+1}^T, \quad (5.26)$$

par conséquent, à partir de (5.20), nous obtenons

$$K_m X_m + X_m J_m = R_m. \tag{5.27}$$

Maintenant nous appliquons [ le Théorème 2.1 dans [104]] à l'équation algébrique de Riccati non symétrique (5.23), se qui termine la démonstration. □

Nous remarquons que $\delta_m = \text{sep}_F(A_m, -D_m)$ est également défini par

$$\delta_m = \inf_{\|P\|=1} \| \mathbb{T}(P) \|_F,$$

où $\mathbb{T}(P) = A_m P + P D_m$.

Les étapes de la méthode que nous proposons (pour résoudre l'équation de Riccati non symétrique de grande taille) sont résumées dans l'algorithme suivant

**Algorithm 10** [La méthode d'Arnoldi étendu par blocs pour résoudre NAREs]

- **Entrées** : les matrices $A$, $D$, $C_1$, $C_2$, $E$ et $F$. Le nombre maximum d'itérations $m_{max}$, et les tolérances *toler* et *dtol*.
- **Sorties** : les matrices $\mathcal{Z}_m^{(1)}$ et $(\mathcal{Z}_m^{(2)})$ de telle sorte que la solution approchée est donnée par
$$X_m \approx \mathcal{Z}_m^{(1)} \left(\mathcal{Z}_m^{(2)}\right)^T.$$
- **Pour** $m = 1, \ldots, m_{max}$
  - Appliquer l'algorithme d'Arnoldi étendu par blocs à $(A, E)$ et $(D^T, F)$ pour obtenir les matrices orthogonales $\mathbb{V}_m$, $\mathbb{W}_m$ et les matrices de Hessenberg par blocs $\mathbb{T}_m^A$ et $\mathbb{T}_m^D$.
  - Résoudre l'équation NAREs de taille réduite
$$\mathbb{T}_m^A Y_m + Y_m (\mathbb{T}_m^D)^T - Y_m C_{1,m} C_{2,m}^T Y_m - \widetilde{E}_m \widetilde{F}_m^T = 0,$$
- **Si** $\| \mathcal{R}_m \|_F <$ *toler*, Arrêt
- **Fin pour**.

Dans la section suivante, nous allons appliquer les méthodes itératives proposées pour la résolution d'un particulier de l'équation algébrique de Riccati utilisée dans la théorie de transport.

## 5.4 Applications à la théorie de transport

Dans cette section, on considère le cas particulier de l'équation algébrique de Riccati non symétrique, largement utilisée dans la théorie de transport, voir [24, 40, 73, 92]. Le

problème de transport consiste à résoudre une équation intégrodifférentielle. Après la discrétisation de cette équation intégrodifférentiel, le problème peut être exprimé sous la forme de l'équation matricielle de Riccati non symétrique suivante

$$(\Delta - eq^T)\,X + X(\Gamma - qe^T) - Xqq^TX - ee^T = 0, \tag{5.28}$$

où les matrices et les vecteurs ci-dessus dépendent des paramètres $c$ et $\alpha$ satisfaisant $0 < c \leq 1$, $0 \leq \alpha < 1$, pour plus de détails voir [24, 38, 40, 73, 92]. Cette équation particulière est de la forme LrNARE donnée à (5.14) avec les matrices suivantes

$$A = \Delta - eq^T,\ D = \Gamma - qe^T,\ C_1 = C_2 = q, \text{ et } E = F = e. \tag{5.29}$$

Les matrices $\Delta$ et $\Gamma$ dans l'équation NARE (5.28) sont données par

$$\Delta = \mathtt{diag}(\delta_1, \ldots, \delta_n); \quad \Gamma = \mathtt{diag}(\gamma_1, \ldots, \gamma_n), \tag{5.30}$$

où

$$\delta_i = \frac{1}{cx_i(1-\alpha)}, \text{ et } \gamma_i = \frac{1}{cx_i(1+\alpha)}, i = 1, \ldots, n. \tag{5.31}$$

Les vecteurs $e$ et $q$ sont donnés par

$$e = (1, \ldots, 1)^T, \ q = (q_1, \ldots, q_n)^T \text{ avec } q_i = \frac{w_i}{2x_i}, i = 1, \ldots, n. \tag{5.32}$$

Nous appliquons ensuite la méthode d'Arnoldi étendue à l'équation NARE (5.28) pour obtenir des solutions approximatives bas de classement. Nous remarquons que l'application de la méthode ci-dessus, nous utilisons des opérations « matrice-vecteur» de la forme $A^{-1}v$ et $D^{-T}v$. Comme les matrices $A$ et $D$ impliquées dans l'équation algébrique de Riccati non symétrique (5.28) sont la somme d'une matrice diagonale et d'une matrice de petit rang, nous pouvons utiliser la formule Sherman-Morrison-Woodbury pour calculer $A^{-1}u$ et $D^{-1}u$, où $u \in \mathbb{R}^n$.
Donc nous avons :

$$A^{-1}u = (\Delta - eq^T)^{-1}u = \Delta^{-1}u + \frac{\Delta^{-1}e\, q^T\, \Delta^{-1}u}{1 - q^T\, \Delta^{-1}e}, \tag{5.33}$$

et

$$D^{-1}u = (\Gamma - qe^T)(\Delta - eq^T)^{-1}u = \Gamma^{-1}u + \frac{\Gamma^{-1}q\, e^T\, \Gamma^{-1}u}{1 - e^T\, \Gamma^{-1}q}. \tag{5.34}$$

Cette formule permet de réduire le temps de calcul et permet également de réduire la place en mémoire.

## 5.5 Résultats numériques

Dans cette section, nous donnons quelques exemples numériques de l'équation de Riccati non symétrique NAREs pour montrer l'efficacité des méthodes proposées. Les différentes expériences numériques ont été réalisées sur un ordinateur Portable d'Intel Core i5 à 1,6 GHz et 4G de RAM. Les algorithmes ont été codés dans Matlab2009. Nous avons comparé la performance de la méthode d'Arnoldi étendu par blocs et la méthode Newton-Block-Arnoldi avec la méthode RRE (Reduced Rank Extrapolation) proposée par El-Moallem et Sadok pour résoudre l'équation de Riccati non symétrique, pour plus de détails voir [38]. Pour l'algorithme d'Arnoldi étendu par blocs, le critère d'arrêt était

$$\|\mathcal{R}(X_m)\|_F \,/\, \|E\,F^T\|_F < 10^{-11},$$

où la norme de $\|\mathcal{R}(X_m)\|$ résiduelle a été calculée en utilisant le théorème 5.3.1. Pour la méthode de Newton-BA et la méthode RRE, les itérations étaient arrêtée lorsque

$$\| X_{k+1} - X_k \|_F \,/\|X_k\|_F < 10^{-11}.$$

Nous considérons le problème venant de la théorie de transport comme expliqué dans la section 5. Nous avons utilisé différentes valeurs des paramètres $\alpha$ et $c$, et également différent tailles du problème de transport. Le tableau (**??**) et le tableau (3.2) rapportent sur les résultats obtenus par la méthode d'Arnoldi étendu par blocs pour résoudre NAREs (algorithme 2) notée (EBA-NARE), la méthode d'Arnoldi Newton-Bloc (Newton-BA) résumée dans l'algorithme 3 pour laquelle l'équation de Sylvester, apparaissant dans chaque étape de l'itération de Newton, a été résolu de manière itérative par l'algorithme d'Arnoldi par blocs et la méthode RRE [38]. Pour la méthode EBA-NARE, l'équation de Riccati non symétrique projetée NARE (5.17) a été résolue en utilisant le programme `sda_affine_mnare` de Bini, Iannazzo et Meini pour plus de détails voir[24].

**Exemple 1.**

Dans ce premier exemple, nous avons choisi $c = 0.5$ et $\alpha = 0.5$. Nous avons d'abord calculé les approximations $X_N$ et $X_E$ données par le Newton-BA et par l'EBA-NARE pour le taille $n = 2000$ et obtenir le suivant norme d'erreur $\| X_N - X_E \|_F = 1.2 \cdot 10^{-10}$ cette résultat signifie que la solution approchée $X_E$ de la méthode EBA-NARE obtenue se rapproche de la solution non négative minimale $X_N$ obtenue par la méthode de Newton.

Nous considérons maintenant des problèmes avec les tailles suivantes $n = 4000$, $n = 10000$, $n = 20000$ et $n = 36000$. Dans le tableau (5.1), une liste des normes résiduelles relatives obtenues pour chaque méthode et le temps de calcul correspondant (en secondes).

Pour toutes les expériences, les itérations de la méthode de Newton ne dépassent pas 5 itérations. Le nombre maximum d'itérations intérieures était $itermax = 50$ et ces itérations intérieures ont également été arrêtée lorsque le résidu correspondant est inférieure à $tol = 10^{-10}$. Nous remarquons que les approximations $X_m$ produite par l'algorithme EBA-NARE sont données sous la forme factorisée (5.21). Ces approximations sont numériquement de petit rang, par exemple, lorsque $n = 4000$, le rang de l'approximation obtenue par la méthode EBA-NARE était égale à 22, et que le rang correspondant à l'approximation obtenue par l'approche de Newton-BA était 23.

TABLE 5.1: Résultats pour les exemples de NAREs dans la théorie de transport avec $c = 0.5$ et $\alpha = 0.5$.

| | Méthode EBA-NARE | | Méthode Newton-BA | | Méthode RRE | |
|---|---|---|---|---|---|---|
| $n$ | Res. Norm | CPU time | Res. Norm | CPU time | Res. Norm | CPU time |
| 4000 | $2.7 \cdot 10^{-12}$ | **1.1s** | $3.5 \cdot 10^{-12}$ | $7.5s$ | $2.5 \cdot 10^{-12}$ | $6.7s$ |
| 10000 | $1.5 \cdot 10^{-12}$ | **2.5s** | $4.5 \cdot 10^{-12}$ | $34s$ | $1.7 \cdot 10^{-12}$ | $34.5s$ |
| 20000 | $5.3 \cdot 10^{-12}$ | **6.5s** | $6.5 \cdot 10^{-12}$ | $415s$ | $6.5 \cdot 10^{-12}$ | $790s$ |
| 36000 | $9.2 \cdot 6.2^{-12}$ | **194s** | $5.2 \cdot 10^{-12}$ | $1430s$ | $---$ | $> 3000s$ |

**Exemple 2.**

Dans le deuxième exemple, nous considérons également l'équation matricielle de Riccati non symétrique dans la théorie de transport avec $c = 0.9999$ et $\alpha = 10^{-8}$. Cet exemple correspond à un cas difficile à résoudre par les méthodes itératives (le problème lié à la singularité de la matrice Jacobienne) pour plus de détails voir [38, 49, 52]. Dans le tableau 5.2, nous avons présenté les normes des résidus et les temps de calcul CPU obtenu pour les trois méthodes : (EBA-NARE), (Newton-BA) et méthode RRE. Un maximum de $kmax = 30$ itérations a été autorisé aux itérations Newton extérieures. ici aussi, nous avons utilisé les mêmes critères d'arrêt que dans l'exemple 1. Comme on le voit à partir du Tableau 5.2, les temps de calcul pour la méthode de Newton sont beaucoup plus élevés que ceux obtenus dans le tableau 5.1. Une possibilité pour éviter la difficulté « le problème de singularité de la matrice Jacobienne » de la méthode Newton dans ce cas est l'utilisation des techniques de décalage « shift technique», en transformant l'équation de départ avec une technique de «shift», qui permet d'éviter le problème de singularité pour la matrice Jacobienne. Pour plus de détails voir [38, 49, 52].

TABLE 5.2: Résultats pour les exemples de NAREs dans la théorie de transport avec $c = 0.9999$ et $\alpha = 10^{-8}$.

| | Méthode EBA-NARE | | Méthode Newton-BA | | Méthode RRE | |
|---|---|---|---|---|---|---|
| $n$ | Res.Norm | CPU time | Res.Norm | CPU time | Res.Norm | CPU time |
| 4000 | $1.7 \cdot 10^{-12}$ | **1.6s** | $3.5 \cdot 10^{-11}$ | $35s$ | $1.5 \cdot 10^{-12}$ | $13.5s$ |
| 10000 | $2.5 \cdot 10^{-12}$ | **2.7s** | $4.5 \cdot 10^{-11}$ | $125s$ | $2.3 \cdot 10^{-12}$ | $72s$ |
| 20000 | $7.6 \cdot 10^{-12}$ | **7.4s** | $1.5 \cdot 10^{-11}$ | $1950s$ | $4.4 \cdot 10^{-12}$ | $2140s$ |
| 36000 | $7.3 \cdot 6.2^{-12}$ | **211s** | $---$ | $> 3000s$ | $---$ | $> 3000s$ |
| 120000 | $5.2 \cdot 6.2^{-12}$ | **763s** | $---$ | $> 3000s$ | $---$ | $> 3000s$ |

**Exemple 3.** Dans ce troisième exemple, nous considérons l'équation de Riccati non symétrique (LrNARE) donné en (5.12). Les matrices $A$ et $D$ sont obtenues à partir de la discrétisation par différences finie centrées des opérateurs

$$LA_u = \Delta u - e^{xy}\frac{\partial u}{\partial x} + \sin(xy)\frac{\partial u}{\partial y} + 100x, \tag{5.35}$$

et

$$LD_u = \Delta u - x^2 y\frac{\partial u}{\partial x} + x\cos(xy)\frac{\partial u}{\partial y} + 10, \tag{5.36}$$

respectivement, sur le carré unité $[0,1]\times[0,1]$ avec des conditions aux bord de Dirichlet homogènes. Le nombre de points de la grille intérieure dans chaque direction a été $n_0$ pour l'opérateur $LA_u$ et $p_0$ pour l'opérateur $LD_u$. La dimension des matrices $A$ et $C$ sont $n = n_0^2$ et $p = p_0^2$ respectivement. Différents choix de valeurs ont été faits pour $n_0$ et $m_0$. Les matrices $E$, $F$, $C_1$ et $C_2$ sont des matrices aléatoires uniformément distribuées dans l'intervalle $[0,\ 1]$ et $s = 5$. Nous utilisé le même critère d'arrêt comme pour les exemples premiers.

Le tableau 5.3 donne les résultats obtenus par les trois méthodes EBA-NARE, Newton-BA et la méthode SDA [24, 50]. Pour les trois méthodes, nous avons reporté la norme de Frobenius du résidu et le temps CPU (en secondes) de calcul. Comme indiqué sur le tableau 5.3, les méthodes proposées donne les meilleurs résultats en terme de temps de calcul.

Table 5.3: Résultats pour les exemples 4, comparaisons avec la méthode SDA.

| | Méthode EBA-NARE | | Méthode Newton-BA | | Méthode SDA | |
|---|---|---|---|---|---|---|
| $n,\ p$ | Res.Norm | CPU time | Res.Norm | CPU time | Res.Norm | CPU time |
| $n = 400,$ $p = 225.$ | $3.7 \cdot 10^{-12}$ | **0.6s** | $6.5 \cdot 10^{-12}$ | $1.8s$ | $2.5 \cdot 10^{-12}$ | $4.7s$ |
| $n = 40000,$ $p = 22500$ | $2.4 \cdot 6.2^{-12}$ | **82.5s** | $4.5 \cdot 10^{-12}$ | $124s$ | $-\ -\ -\ -\ -$ | |
| $n = 122500$ $p = 90000$ | $6.3 \cdot 6.2^{-12}$ | **240s** | $6.5 \cdot 10^{-12}$ | $415s$ | $-\ -\ -\ -$ | |

## 5.6 Conclusion

Nous avons présenté dans ce chapitre, deux nouvelles méthodes itératives pour calculer la solution approchée de rang inférieur d'équations algébriques de Riccati non symétriques de grande taille. La première basée sur des sous-espace de Krylov étendus par blocs et la condition d'orthogonalité de Galerkin. Dans la deuxième méthode, nous avons présenté la méthode de Newton-bloc Arnoldi, basée sur la méthode de Newton et des sous-espaces de Krylov par blocs : à chaque itération de la méthode de Newton, on résout une équation de Sylvester de grande taille par l'utilisation de l'algorithme d'Arnoldi par blocs. Nous

avons également présenté des résultats de majoration de l'erreur. Nous avons appliqué ces méthodes itératives à l'équation de Riccati non symétrique bien connue dans la théorie du transport. Les résultats numériques montrent que les approches proposées sont efficaces pour les problèmes de grande taille.

# Bibliographie


[1] H. Abou-Kandil, G. Freiling, V. Ionescu, and G. Jank, Matrix Riccati Equations in Control and Systems Theory, Systems and Control : Foundations and Applications, Birkh auser, Boston, 2003.

[2] B. D. O. Anderson, and J. B. Moore, Linear Optimal Control, Prentice-Hall, Englewood Cliffs, 1971.

[3] B. D. O. Anderson, and J. B. Moore, Optimal Control-Linear Quadratic Methods, Prentice-Hall, Englewood Cliffs, NJ, 1990.

[4] A. C. Antoulas, D.C. Sorensen, Projection methods for balanced model reduction, Technical Report, Rice University, Houston, TX, 2001.

[5] A. C. Antoulas, Approximation of Large-Scale Dynamical Systems, SIAM, 2005.

[6] W. F. Arnold and A. J. Laub, Generalized eigenproblem algorithms and software for algebraic Riccati equations. Proc. IEEE, 72, pp : 1746–1754, 1984.

[7] J. Baglama, L. Reichel, D. Richmond, An augmented LSQR method, Numer. Algorithms, 64 (2013), pp. 263–293.

[8] A. Y. Barraud, A numerical algorithm to solve $A^T XA - X = Q$, IEEE Trans. Autom. Contr., AC-22(1977), pp. 883–885.

[9] Z-Z. Bai, Y-H. Gao, L-Z. Lu, Fast iterative schemes for nonsymmetric algebraic Riccati equations arising from transport theory, SIAM J. Sci. Comp., 30(2008), pp. 804–818.

[10] Z-Z. Bai, X-X. Guo, S-F. Xu, Alternately linearized implicit iteration methods for the minimal nonnegative solutions of the nonsymmetric algebraic Riccati equations, Num. Lin. Alg. with App, 13(2006), pp. 655–674.

[11] R.H. Bartels, G.W. Stewart, Solution of the matrix equation $AX + XB = C$, Algorithm 432, Comm. ACM 15(1972), pp. 820–826.

[12] M. Bellalij, K. Jbilou, H. Sadok, New convergence results on the global GMRES method for diagonalizable matrices, Journal of Computational and Applied Mathematics 219 (2008), pp. 350–358.

[13] P. Benner, Factorized solution of Sylvester equations with applications in control, in : B. De Moor, B. Motmans, J. Willems, P. Van Dooren, V. Blondel (Eds.), Proceedings of the Sixteenth International Symposium on Mathematical Theory of Network and Systems, MTNS, Leuven, Belgium, 2004.

[14] P. Benner and R. Byers. An exact line search method for solving generalized continous algebraic Riccati equations. IEEE Trans. Automat. Control, 43, pp : 101–107, (1998).

[15] P. Benner, J. Li, and T. Penzl. Numerical solution of large Lyapunov equations, Riccati equations and linear-quadratic optimal control problems. Numer. Linear Alg. Appl., 15, pp. 755–777, 2008.

[16] P. Benner and H. Fa$\beta$bender. An implicitly restarted symplectic Lanczos method for the hamiltonian eigenvalue problem. Linear Alg. Appl., 263, pp. 75–111, 1997.

[17] A. Bouhamidi , K. Jbilou, Stien implicit Runge-Kutta methods with high stage order for large-scale ordinary differential equations. Applid Numerical Mathematics, 61, pp. 149–159, 2011.

[18] A. Bouhamidi , K. Jbilou, Sylvester Tikhonov-regularization methods in image restoration, J. Comput. Appl. Math., 206(1), pp. 86–98, 2007.

[19] A. Bouhamidi , K. Jbilou, A note on the numerical approximate solutions for generalized Sylvester matrix equations with applications, Applied Mathematics and Computation 206 (2008), pp. 687–694.

[20] S. Bittanti, A. Laub, and J. C. Willems, The Riccati equation, Springer-Verlag, Berlin, 1991.

[21] P. Benner, R.C. Li, N. Truhar, On the ADI method for Sylvester equations, J. of Comput. Appl. Math., (233)2009, pp. 1035–1045.

[22] P. Benner, H. Mena, and J. Saak, On the parameter selection problem in the Newton-ADI iteration for large-scale Riccati equations, Electron. Trans. Numer. Anal., 29 (2008), pp. 136–149.
http ://etna.math.kent.edu/vol.29.2007–2008/pp136–149.dir/

[23] P. Benner and P. Kürschner, Computing real low-rank solutions of Sylvester equations by the factored ADI method, Comput. Math. Appl., 67 (2014), pp. 1656–1672.

[24] D.A. Bini, B. Iannazzo, B. Meini, Numerical Solution of Algebraic Riccati Equations, SIAM, Philadelphia, PA, 2012.

[25] D.A. Bini, B. Iannazzo, G. Latouche, B. Meini, On the solution of algebraic Riccati equations arising in fluid queues, Linear Algebra Appl., 413(2006), pp. 474–494.

[26] D.A. Bini, B. Iannazzo and F. Poloni, A fast Newtons method for a nonsymmetric algebraic Riccati equation, SIAM J. Matrix Anal. Appl., 30(2008), pp : 276–290.

[27] R. Byers, Solving the algebraic Riccati equation with the matrix sign function. Linear Algebra and its Applications, 85, pp : 267–279, 1987.

[28] J-P Chehab, M. Raydan, An implicit preconditioning strategy for large-scale generalized Sylvester equations, Applied Mathematics and Computation, 217(2011), pp : 8793–8803.

[29] A. Bouhamidi, M. Hached, M. Heyouni and K. Jbilou, A preconditioned block Arnoldi for large Sylvester matrix equations, Numer. Lin. Alg. Appl., 2(2013), pp : 208–219.

[30] R. Bouyouli, K. Jbilou, R. Sadaka, H. Sadok, Convergence properties of some block Krylov subspace methods for multiple linear systems, Journal of Computational and Applied Mathematics, 196 (2006), pp. 498–511.

[31] Callier, F.M and Desoer, Linear System Theory, SpringerVerlag (1991).

[32] B. N. Datta, Numerical Methods for Linear Control Systems Design and Analysis, Elsevier Academic Press, 2003.

[33] B.N. Datta, Numerical Methods for Linear Control Systems, Elsevier Academic press, 2004.

[34] B.N. Datta, Krylov-subspace methods for large scale matrix problems in control, Generation of Computer Systems, 19(2003), pp. 125–126.

[35] B.N. Datta, K. Datta, Theoretical and computational aspects of some linear algebra problems in control theory, in :C.I. Byrnes, A. Lindquist (EDs), Computational and Combinatorial Methods in Systems Theory, Elsevier, Amsterdam (1986), pp 201–212.

[36] V. Druskin, L. Knizhnerman, Extended Krylov subspaces : approximation of the matrix square root and related functions, SIAM J. Matrix Anal. Appl. 19 (3) (1998), pp. 755–771.

[37] E. J. C. Doyle, K. Glover, P. P. Khargonekar, and B. A. Francis, State space solutions to standard $H_2$ and $H_\infty$ control problems, IEEE Trans. Automat. Control, 34 (1989), pp. 831–846.

[38] R. El-Moallem and H. Sadok, Vector extrapolation methodsapplied to algebraic Riccati equations arising in transport theory, Elect. Trans. Numer. Anal., 40(2013), pp. 489–506.

[39] A. El Guennouni, K. Jbilou, A.J. Riquet, Block Krylov subspace methods for solving large Sylvester equations, Numer. Alg. 29 (2002), pp.75–96.

[40] B. D. Ganapol, An investigation of a simple transport model, Transport Theory Statist. Phys., 21(1992), pp. 1–37.

[41] C-H. Guo, Nonsymmetric algebraic Riccati equations and Wiener-Hopf factorization for M-matrices, SIAM J. Matrix Anal. Appl., 23(1)(2001), pp. 225–242.

[42] K. Glover, D.J. N. Limebeer, J.C. Doyle, E.M. Kasenally and M.G. Safonov, A characterisation of all solutions to the four block general distance problem, SIAM J. Control Optim., 29(1991), pp. 283–324.

[43] G.H. Golub, and W. Kahan, Calculating the singular values and pseudoinverse of a matrix, SIAM J. Numer. Anal. 2 (1965), pp. 205–224.

[44] G.H. Golub and C.F. Van Loan, Matrix Computations, (Johns Hopkins Studies in Mathematical Sciences)(3rd Edition). The Johns Hopkins University Press,1996.

[45] G.H. Golub, S. Nash, C. Van Loan, A Hessenberg Schur method for the problem $AX + XB = C$ , IEEE Trans. Automat. Control 24 (1979), pp. 909–913.

[46] C. H. Guo, N. J. Higham, Iterative solution of a nonsymmetric algebraic Riccati equation. SIAM, J. Matrix Anal. Appl. 29(2)(2007), pp : 396–412.

[47] C. H. Guo, Nonsymmetric algebraic Riccati equations and Wiener-Hopf factorization for $M$-matrices, SIAM J. Matrix Anal. Appl. 23 (1) (2001), pp 225–242.

[48] C-H. Guo and A. J. Laub, On the iterative solution of a class of nonsymmetric algebraic Riccati equations, SIAM J. Matrix Anal. Appl., 22(2000), pp. 376–391.

[49] C.H. Guo, B. Iannazzo, and B. Meini, On the doubling algorithm for a (shifted) nonsymmetric algebraic Riccati equation, SIAM J. Matrix Anal. Appl., 29(4)(2007), pp : 1083–1100.

[50] X.-X. Guo, W.-W. Lin, and S.-F. Xu, A structure-preserving doubling algorithm for nonsymmetric algebraic Riccati equation, Numer. Math., 103(2006), pp. 393–412.

[51] Guang-Da Hu and Qiao Zhu, Bounds of modulus of eigenvalues based on Stein equation. Kybernetika, 46(4), pp. 655–664, 2010.

[52] C. He, B. Meini, NH. Rhee, A shifted cyclic reduction algorithm for quasi- birth-death problems, SIAM Journal on Matrix Analysis and Applications, 23(3)(2001), pp. 673–691.

[53] S.J. Hammarling, Numerical solution of the stable, nonnegative definite Lyapunov equation, IMA J. Numer. Anal. 2 (1982), pp. 303–323.

[54] R.E. Hartwig, Resultants and the solution of $AX - XB = -C$, SIAM J. Appl. Math. 23 (1) (1972), pp. 104–117.

[55] M. Heyouni, K. Jbilou, An extended Block Arnoldi algorithm for large-scale solutions of the continuous-time algebraic Riccati equation, ETNA 33 (2009), pp. 53–62. http ://etna.mcs.kent.edu/vol.33.2008-2009/pp53-62.dir

[56] M. Heyouni, Extended Arnoldi methods for large low-rank Sylvester matrix equations, Appl. Num. Math., 60(11)(2010), pp. 1171–1182.

[57] D.Y. Hu, L. Reichel, Krylov-subspace methods for the Sylvester equation, Linear Algebra Appl. 172 (1992), pp. 283–313.

[58] M. Herty, R. Pinnau, M. Sead, On Optimal Control Problems in Radiative Transfer, Optimization Methods and Software, 22(2007), pp. 917–936.

[59] Carl Jagels and Lothar Reichel, Recursion relations for the extended Krylov subspace method. Linear Algebra Appl., 434(7)(2011), pp. 1716–1732.

[60] I. M. Jaimoukha and E. M. Kasenally, Krylov subspace methods for solving large Lyapunov equations, SIAM J. Numer. Anal.,31(1994), pp. 227–251.

[61] K. Jbilou, ADI preconditioned Krylov methods for large Lyapunov matrix equations, Linear Algebra and its Applications. 432 (2010) pp. 2473–2485.

[62] K. Jbilou, Low-rank approximate solution to large Sylvester matrix equations, App. Math. Comput., 177(2006), pp. 365–376.

[63] K. Jbilou, Approximate solutions to large Stein matrix equations, World Academy of Science, Engineering and Technology Vol :70(2012), pp : 19–25.

[64] K. Jbilou, Block Krylov subspace methods for large continuous-time algebraic Riccati equations, Num. Alg., 34(2003), pp : 339–353.

[65] K. Jbilou, An Arnoldi based algorithm for large algebraic Riccati equations, Appl. Math. Lett., 19 (2006), pp. 437–444.

[66] K. Jbilou and A. Messaoudi, A Computational Method for Symmetric Stein Matrix Equations, Numerical Linear Algebra in Signals, Systems and Control. Volume 80(2011), pp 295–311.

[67] K. Jbilou, A. Messaoudi, H. Sadok, Global FOM and GMRES algorithms for matrix equations, Appl. Numer. Math. 31(1999), pp. 49–63.

[68] K. Jbilou and A. J. Riquet, Projection methods for large Lyapunov matrix equations, Lin. Alg. Appl, 415(2006), pp. 344–358.

[69] K. Jbilou and H. Sadok, Vector extrapolation methods. Applications and numerical comparison, J. Comput. Appl. Math., 122(2000), pp : 149–165.

[70] K. Jbilou, H. Sadok, Global Lanczos-based methods with applications, Technical Report LMA 42, Univérsité du Littoral, Calais, France, 1997.

[71] Jonq Juang and Wen-Wei Lin, nonsymmetric algebric Riccati equations and Hamiltonian-Like matrices. SIAM J. MATRIX ANAL. APPL. Vol. 20 (1998), pp. 228–243.

[72] J. Juang, Existence of algebraic matrix Riccati equations arising in transport theory, Lin. Alg. Appl., 230(1995), pp : 89–100.

[73] J. Juang and D. Chen, Iterative solution for a certain class of algebraic matrix Riccati equations arising in transport theory, Transp. Theor. Statis. Phys., 22(1), pp : 65–80.

[74] Z. Jia and Y. Sun, A $QR$ decomposition based solver for the least squares problems from the minimal residual method for the Sylvester equation, J. Comput. Math., 25(2007), pp. 531–542.

[75] S. Karimi, B. Zali, The block preconditioned LSQR and GL-LSQR algorithms for the block partitioned matrices, Applied Mathematics and Computation, 227(2014), pp. 811–820.

[76] A. Klein and P.J.C. Spreij, On Stein's equation, Vandermonde matrices and Fisher's information matrix of time series processes. Part I : The autoregressive moving average process. Lin. Alg. Appl., 329(13), pp. 9–47, 2001.

[77] André Klein and Peter Spreij. On the solution of Stein's equation and Fisher's information matrix of an ARMAX process. Linear Algebra and its Applications, 396(2005), pp :1–34.

[78] D. Kressner and C. Tobler, Low-rank tensor Krylov subspace methods for parametrized linear systems, SIAM J. Matrix Anal. Appl., 32 (2011), pp. 1288–1316.

[79] D.L. Kleinman, On an iterative technique for Riccati equation computations, IEEC Trans. Autom. Contr., 13(1968), pp : 114–115.

[80] P. Lancaster, and L. Rodman, Algebraic Riccati Equations, Clarendon Press, Oxford, (1995).

[81] P. Lancaster and M. Tismenetsky, The Theory of Matrices, Academic Press, Orlando, 1985.

[82] Y. Lin and V. Simoncini, Minimal residual methods for large scale Lyapunov equations, App. Num. Math., 72(2013), pp : 52–71.

[83] P. Lancaster, L. Rodman, The Algebraic Riccati Equations, Clarendon Press, Oxford, 1995.

[84] Y. Lin, I. Bao and Y. Wei, A modified Newton method for solving non-symmetric algebraic Riccati equations arising in transport theory, IMA J. Numer. Anal., 29(2008), pp : 215–224.

[85] L.-Z. Lu, Newton iterations for a non-symmetric algebraic Riccati equation, Numer. Lin. Alg. Appl., 12(2005), pp : 191–200.

[86] L.-Z. Lu, Solution form and simple iteration of a nonsymmetric algebraic Riccati equation arising in transport theory, SIAM J. Matrix Anal. Appl., 26(2005), pp : 679–685.

[87] J. Lasalle, S. Lefschetz, Stability of Lyapunov Direct Methods, Academic Press, New York, 1961.

[88] A. J. Laub, A Schur method for solving algebraic Riccati equations, IEEE Trans. Automat. Control, 24 (1979), pp. 913–921.

[89] T. Penzl, LYAPACK : A MATLAB toolbox for large Lyapunov and Riccati equations, model reduction problems, and linear-quadratic optimal control problems, software available at
https ://www.tu-chemnitz.de/sfb393/lyapack/.

[90] C. Moler and C. Van loan, Ninteen dubious ways to compute the exponential of a matrix, SIAM Rev., 20(1978), pp. 801–836.

[91] V. Mehrmann. The Autonomous Linear Quadratic Problem, Theory and Numerical Solution. Lecture Notes in Control and Information Sciences, Vol. 63, Springer, Heidelberg, (1995).

[92] V. Mehrmann and H. Xu, Explicit solutions for a Riccati equations from transport theory, SIAM J. Matrix Anal., 30(2008), pp : 1339–1357.

[93] V. Mehrmann, H. Xu, Explicit solutions for a Riccati equation from transport theory, SIAM J. Matrix Anal. Appl. 30 (4)(2008), pp : 1339–1357.

[94] C. C. Paige and M. A. Saunders, Solution of sparse indefinite systems of linear equations. SIAM J. Numer. Anal., 12(4)(1975), pp : 617–629.

[95] C.C. Paige, A. Saunders, LSQR : an algorithm for sparse linear equations and sparse least squares, ACM Trans. Math. Software 8 (1982), pp. 43–71.

[96] Frédéric Rotella, Pierre Borne, Théorie et pratique du calcul matriciel, 1995.

[97] L. C. G. Rogers, Fluid models in queueing theory and Wiener-Hopf factorization of Markov chains, Ann. Appl. Probab., 4(2)(1994), pp : 390–413.

[98] Y. Saad, Numerical solution of large Lyapunov equations, in Signal Processing, Scattering, Operator Theory and Numerical Methods, M.A. Kaashoek, J.H. Van Shuppen and A.C. Ran, eds., Birkhaser, Boston, 1990, pp. 503–511.

[99] Y. Saad and M. H. Schultz, GMRES : A generalized minimal residual algorithm for solving nonsymmetric linear systems, SIAM J. Sci. Statis. Comput., 7 (1986), pp. 856–869.

[100] Y. Saad, Iterative Methods for Sparse Linear Systems, Society for Industrial and Applied Mathematics 2éme edition. SIAM, Philadelpha, PA, 2003.

[101] M. Sadkane, Block Arnoldi and Davidson methods for unsymmetric large eigenvalue problems, Numer. Math., 64(1993), pp. 687–706.

[102] V. Simoncini, A new iterative method for solving large-scale Lyapunov matrix equations, SIAM J. Sci. Comput, 29(2007), pp. 1268–1288.

[103] E. de Souza, S.P. Bhattacharyya, Contollability, observability and solution of $AX - XB = C$, Linear Algebra Appl. 39 (1981), pp. 167–188.

[104] G.W. Stewart and J.G. Sun, Matrix Perturbation Theory, Academic Press, New York, (1990).

[105] R.A. Smith, Matrix calculations for Liapunov quadratic forms, J. Different. Eqs. 2(1966), pp. 208–217.

[106] F. Toutounian and S. Karimi, Global least squares method (Gl-LSQR) for solving general linear systems with several right-hand sides, Applied Mathematics and Computation 178 (2006), pp.452–460.

[107] P. Van Dooren, A generalized eigenvalue approach for solving Riccati equations. SIAM J. Sci. Statist. Comput., 2, pp :121–135, (1981).

[108] P. Van Dooren, Gramian based model reduction of large-scale dynamical systems, in Numerical Analysis, Chapman and Hall, pp. 231–247, CRC Press London, 2000.

[109] W.M. Wonham, On a matrix Riccati equation of stochastic control. SIAM J. Contr., 6, pp : 681–697, 1968.

[110] D. Williams, A potential-theoretical note on the quadratic Wiener-Hopf equation for $Q$-matrices, in Seminar on Probability XVI, Lecture Notes in Mathematics 920, pp. 91–94, Springer-Verlag, Berlin, 1982.

[111] Zhou, K., Doyle, J.C. and Glover, Robust Optimal Control. Prentice Hall, New-Jersey (1995).

# Résumé

Nous nous intéressons dans cette thèse, à l'étude des méthodes itératives pour la résolution d'équations matricielles de grande taille : Lyapunov, Sylvester, Riccati et Riccati non symétrique.

L'objectif est de chercher des méthodes itératives plus efficaces et plus rapides pour résoudre les équations matricielles de grande taille. Nous proposons des méthodes itératives de type projection sur des sous espaces de Krylov par blocs

$$\mathbb{K}_m(A, V) = Image\{V, AV, \ldots, A^{m-1}V\},$$

ou des sous espaces de Krylov étendus par blocs

$$\mathcal{K}_m^e(A, V) = Image\{V, A^{-1}V, AV, A^{-2}V, A^2V, \cdots, A^{m-1}V, A^{-m+1}V\}$$

Ces méthodes sont généralement plus efficaces et rapides pour les problèmes de grande dimension.

Nous avons traité d'abord la résolution numérique des équations matricielles linéaires : Lyapunov, Sylvester et Stein. Nous avons proposé une nouvelle méthode itérative basée sur la minimisation de résidu MR et la projection sur des sous espaces de Krylov étendus par blocs $\mathcal{K}_m^e(A, V)$. L'algorithme d'Arnoldi étendu par blocs permet de donner un problème de minimisation projeté de petit taille. Le problème de minimisation de taille réduit est résolu par différentes méthodes directes ou itératives. Nous avons présenté aussi la méthode de minimisation de résidu basée sur l'approche global à la place de l'approche bloc. Nous projetons sur des sous espaces de Krylov étendus Global

$$K_m^e(A, V) = sev\{V, A^{-1}V, AV, A^{-2}V, A^2V, \cdots, A^{m-1}V, A^{-m+1}V\}.$$

Nous nous sommes intéresses en deuxième lieu à des équations matricielles non linéaires, et tout particulièrement l'équation matricielle de Riccati dans le cas continu et dans le cas non symétrique appliquée dans les problèmes de transport. Nous avons utilisé la méthode de Newton et l'algorithme MINRES pour résoudre le problème de minimisation projeté. Enfin, nous avons proposé deux nouvelles méthodes itératives pour résoudre les équation de Riccati non symétriques de grande taille : la première basée sur l'algorithme d'Arnoldi étendu par bloc et la condition d'orthogonalité de Galerkin, la deuxième est

de type Newton-Krylov, basée sur la méthode de Newton et la résolution d'une équation de Sylvester de grande taille par une méthode de type Krylov par blocs. Pour toutes ces méthodes, les approximations sont données sous la forme factorisée, ce qui nous permet d'économiser la place mémoire en programmation. Nous avons donné des exemples numériques qui montrent bien l'efficacité des méthodes proposées dans le cas de grandes tailles.

**Mots clés :** Sous espaces de Krylov étendu, méthodes blocs et globales, Approximation de rang inférieur, Lyapunov, Sylvester, Riccati continu, Riccati non symétrique, Méthode de Newton, Théorie de transport, Condition de Galerkin, Méthode de minimisation de résidu.

# Abstract

In this thesis, we focus in the studying of some iterative methods for solving large matrix equations such as Lyapunov, Sylvester, Riccati and nonsymmetric algebraic Riccati equation. We look for the most efficient and faster iterative methods for solving large matrix equations. We propose iterative methods such as projection on block Krylov subspaces

$$\mathbb{K}_m(A, V) = Range\{V, AV, \ldots, A^{m-1}V\},$$

or block extended Krylov subspaces

$$\mathcal{K}_m^e(A, V) = Range\{V, A^{-1}V, AV, A^{-2}V, A^2V, \cdots, A^{m-1}V, A^{-m+1}V\}.$$

These methods are generally most efficient and faster for large problems.

We first treat the numerical solution of the following linear matrix equations : Lyapunov, Sylvester and Stein matrix equations. We have proposed a new iterative method based on Minimal Residual MR and projection on block extended Krylov subspaces $\mathcal{K}_m^e(A, V)$. The extended block Arnoldi algorithm gives a projected minimization problem of small size. The reduced size of the minimization problem is solved by direct or iterative methods. We also introduced the Minimal Residual method based on the global approach instead of the block approach. We projected on the global extended Krylov subspace

$$K_m^e(A, V) = Span\{V, A^{-1}V, AV, A^{-2}V, A^2V, \cdots, A^{m-1}V, A^{-m+1}V\}.$$

Secondly, we focus on nonlinear matrix equations, especially the matrix Riccati equation in the continuous case and the nonsymmetric case applied in transportation problems. We used the Newton method and MINRES algorithm to solve the projected minimization problem. Finally, we proposed two new iterative methods for solving large nonsymmetric Riccati equation : the first based on the algorithm of extended block Arnoldi and Galerkin condition, the second type is Newton-Krylov, based on Newton's method and the resolution of the large matrix Sylvester equation by using block Krylov method.

For all these methods, approximations are given in low rank form, wich allow us to save memory space. We have given numerical examples that show the effectiveness of the methods proposed in the case of large sizes.

**Keywords :** Extended Krylov subspaces, Low-rank approximation, Lyapunov equation, Matrix Sylvester equation, Riccati equation, Nonsymmetric Riccati equation, Transport theory, Newton method, minimal residual method MR, Galerkin condition.