



HAL
open science

Méthodes de sous-espaces de Krylov matriciels appliquées aux équations aux dérivées partielles

Mustapha Hached

► **To cite this version:**

Mustapha Hached. Méthodes de sous-espaces de Krylov matriciels appliquées aux équations aux dérivées partielles. Mathématiques générales [math.GM]. Université du Littoral Côte d'Opale, 2012. Français. NNT : 2012DUNK0315 . tel-00919796

HAL Id: tel-00919796

<https://theses.hal.science/tel-00919796>

Submitted on 17 Dec 2013

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

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

UNIVERSITÉ DU LITTORAL CÔTE D'OPALE (U.L.C.O)
ACADÉMIE DE LILLE
ÉCOLE DOCTORALE RÉGIONALE SCIENCES POUR L'INGÉNIEUR N°072

ORDRE n°

THÈSE DE DOCTORAT

Discipline : Mathématiques appliquées

présentée par

Mustapha HACHED

**Méthodes de sous-espaces de Krylov matriciels
appliquées aux équations aux dérivées partielles.**

Soutenue publiquement le 7 décembre 2012 devant la
commission d'examen composée de

jury

M. Gérard MEURANT	Docteur d'Etat, CEA	rapporteur
M. Miloud SADKANE	Professeur, Université de Brest	rapporteur
M. Hassane SADOK	Professeur, ULCO	examineur
M. Christian GOUT	Professeur, INSA de Rouen	examineur
M. Mohammed SEID	Professeur, Durham University	examineur
M. Abderrahman BOUHAMIDI	Maître de conférences HDR, ULCO	directeur
M. Khalide JBILOU	Professeur, ULCO	directeur

Laboratoire de Mathématiques
Pures et Appliquées Joseph Liou-
ville
Maison de la Recherche Blaise
Pascal
50, rue Ferdinand Buisson
BP 699, 62228 Calais cedex, France

École doctorale S.P.I. 072 ULCO

A Ana, Adil et Amin.

Remerciements

Cette thèse a été préparée au Laboratoire de Mathématiques Pures et Appliquées (LMPA) de l'Université du Littoral Côte d'Opale. Mes premiers remerciements vont à mes deux directeurs de thèse : Abderrahman Bouhamidi et Khalide Jbilou. C'est après un long questionnement que j'ai décidé d'entreprendre cette thèse que l'on pourra qualifier au choix de tardive ou alors de projet bien mûri, cette dernière proposition, plus flatteuse, ayant ma préférence. C'est avec une grande gentillesse qu'Abderrahman Bouhamidi et Khalide Jbilou m'ont exposé leurs travaux et proposé un sujet de thèse portant sur les équations matricielles et leurs applications pour la résolution d'équations aux dérivées partielles par des méthode sans maillage. J'avoue aujourd'hui qu'à ce moment, onze ans après avoir obtenu mon DEA qui portait sur les variétés algébriques et symplectiques, ce sujet me parût très ambitieux. Je les remercie pour leur patience, leur rigueur et leur générosité qui m'ont permis de découvrir et d'aimer le monde de la recherche en mathématiques appliquées.

Je remercie Monsieur Hassane Sadok, directeur du LMPA, pour son accueil, sa gentillesse et ses encouragements.

J'adresse toute ma gratitude à Messieurs Gérard Meurant et Miloud Sadkane pour avoir accepté d'être rapporteurs de ma thèse ainsi qu'à Messieurs Christian Gout et Mohammed Seaid pour bien vouloir faire partie du jury.

J'adresse tous mes remerciements à mes collègues du département chimie de l'IUT A de Lille 1 pour leurs encouragements et leur amitié sans faille. Je ne vous nomme pas, je risquerais d'en oublier...et maintenant que j'ai fini ma thèse, vous ne pardonneriez plus cette distraction dont j'ai fait preuve ces derniers temps!

Durant la dernière année de préparation de ce travail, l'Université des Sciences et Techniques de Lille, m'a octroyé une décharge d'enseignement qui m'a permis de terminer mes travaux dans de bonnes conditions. Je remercie en particulier Messieurs Ahmed Mazzah, chef du département chimie de l'IUT A, Moulay Driss Benchiboun, directeur de l'IUT A et Philippe Rollet, président de l'Université, pour avoir donné un avis favorable à ma demande.

Je remercie mes amis et mes parents qui m'ont encouragé et se sont intéressés à mon projet.

Bien que très stimulante et gratifiante, la préparation d'une thèse, tout en assurant ses enseignements à plein temps, sauf la dernière année, est une tâche très exigeante. S'y ajoutent toutes les contingences de la vie quotidienne. On y sacrifie forcément quelque chose. Je remercie mon épouse Ana et mes enfants Adil et Amin pour leur patience et leur soutien inconditionnel. ¡Muchísimas gracias por haberme permitido lograr mi sueño y... perdonad las molestias!

Résumé

Cette thèse porte sur des méthodes de résolution d'équations matricielles appliquées à la résolution numérique d'équations aux dérivées partielles ou des problèmes de contrôle linéaire.

On s'intéresse en premier lieu à des équations matricielles linéaires. Après avoir donné un aperçu des méthodes classiques employées pour les équations de Sylvester et de Lyapunov, on s'intéresse au cas d'équations linéaires générales de la forme $\mathcal{M}(X) = C$, où \mathcal{M} est un opérateur linéaire matriciel. On expose la méthode de GMRES globale qui s'avère particulièrement utile dans le cas où $\mathcal{M}(X)$ ne peut s'exprimer comme un polynôme du premier degré en X à coefficients matriciels, ce qui est le cas dans certains problèmes de résolution numérique d'équations aux dérivées partielles. On se penche ensuite sur un cas particulier des équations de Sylvester $AX + XB = C$. Dans certaines situations, notamment tirées de problèmes de contrôle linéaire, les matrices A et B sont creuses, de grande taille et la matrice C peut s'écrire sous la forme d'un produit EF^T où E et F sont de petit rang. Nous proposons une approche, notée LR-BA-ADI (Low Rank Block Arnoldi ADI) consistant à utiliser un préconditionnement de type ADI qui transforme l'équation de Sylvester en une équation de Stein que nous résolvons par une méthode de Krylov par blocs. Les itérés sont donnés sous forme factorisée, nous permettant d'économiser de la place mémoire. La performance de cette approche au regard d'autres méthodes est confirmée par des tests comparatifs. Enfin, nous proposons une méthode de type Newton-Krylov par blocs avec préconditionnement ADI pour les équations de Riccati issues de problèmes de contrôle linéaire quadratique. Cette méthode est dérivée de la méthode LR-BA-ADI. Des résultats de convergence et de majoration de l'erreur sont donnés.

Dans la seconde partie de ce travail, nous appliquons les méthodes exposées dans la première partie de ce travail à des problèmes d'équations aux dérivées partielles. Nous nous intéressons d'abord à la résolution numérique d'équations de type Burgers de la forme $\partial_t u + \mu(u \cdot \nabla)u - \nu Lu = f$ sur un ensemble $\Omega \times [t_0, T]$, où Ω est un domaine de \mathbb{R}^d , $d \geq 2$, μ et ν sont deux paramètres et L est un opérateur différentiel linéaire. Dans le cas où Ω est un rectangle de \mathbb{R}^2 , on applique un schéma

de discrétisation en espace par différences finies. On aboutit à un système différentiel non linéaire que nous résolvons par un schéma de Runge-Kutta implicite. Chaque itération donne lieu à la résolution d'une équation matricielle de Stein non symétrique qui sera menée à bien par l'utilisation de la méthode GMRES globale. Ensuite, nous nous intéressons au cas où le domaine borné Ω est choisi quelconque dans \mathbb{R}^d , $d \geq 2$. L'approche choisie repose sur l'utilisation de fonctions à base radiale dans le cadre d'une méthode sans maillage (meshless). Nous proposons un formalisme différent de ce qui est habituellement donné dans la littérature. Nous établissons des résultats théoriques de l'existence de tels interpolants faisant appel à des techniques d'algèbre linéaire.

Les cas stationnaire et évolutif sont traités successivement. Dans le premier cas, par interpolation par des fonctions à base radiale, on se ramène à la résolution d'une équation matricielle non linéaire $R(X) = 0$. La méthode de Newton-inexacte que nous employons demande, à chaque pas, la résolution d'une équation linéaire matricielle de la forme $DR(X) = C$, où $DR(X)$ est la dérivée de Fréchet de R , qui ne peut être identifiée à un polynôme en X à coefficients matriciels, nous amenant alors à utiliser la méthode GMRES globale. Dans le second cas, on se ramène à la résolution d'une équation différentielle ordinaire matricielle non linéaire. On adapte le formalisme de la méthode de Runge-Kutta implicite au cas matriciel et comme dans le cas stationnaire, on se ramène à un problème de résolution d'une équation matricielle non linéaire qui sera traitée de la même façon. Enfin, nous nous intéressons à un problème de contrôle linéaire quadratique. En particulier, nous nous penchons sur l'équation de la chaleur en deux dimensions. La recherche du contrôle qui minimise une certaine énergie nous ramène à la résolution d'une équation de Riccati. Cette équation est résolue numériquement par la méthode de type Newton-Krylov par blocs avec préconditionnement ADI.

Mots-clefs

Approximation, Arnoldi, Burgers, Chaleur, EDP, GMRES, Krylov, Lyapunov, Meshless, Newton, RBF, Riccati, Sylvester.

Abstract

This thesis deals with some matrix equations involved in numerical resolution of partial differential equations and linear control.

We first consider some numerical resolution techniques of linear matrix equa-

tion. After giving a review on classical methods used for Sylvester and Lyapunov equations, we consider the general linear matrix equation of the form $\mathcal{M}(X) = C$, where \mathcal{M} is a linear matrix operator. The global-GMRES method is particularly adapted to the case where $\mathcal{M}(X)$ cannot be expressed as a linear polynomial of the variable X with matricial coefficients. This situation occurs in some problems related to partial differential equations. We then consider a particular case of the Sylvester equation $AX + XB = C$. In some applications, for instance in linear control theory, matrices A and B are large and sparse and the right hand side C can be factorised as EF^T , where E and F are low rank matrices. We propose a new approach, denoted LR-BA-ADI (Low Rank Block Arnoldi ADI) that combines an ADI-type preconditioning technique transforming our equation into a Stein equation. This Stein equation is then numerically solved using a block-Krylov method. The iterates are given under a factored form which allows us to reduce the cost in terms of memory. The performance of this approach with regard to other methods is confirmed by comparative tests. Eventually, we derive from the LR-BA-ADI method a Newton-Krylov algorithm for the numerical resolution of a Riccati equation related to linear quadratic control problems. Some results regarding convergence and error bounds are provided. In the second part of this thesis, we apply these resolution techniques to problems related to partial differential equations. We consider Burgers' type equations : $\partial_t u + \mu(u \cdot \nabla)u - \nu Lu = f$ on $\Omega \times [t_0, T]$, where Ω is a bounded domain of \mathbb{R}^d , $d \geq 2$, μ and ν are two scalar parameters. In the case where Ω is a rectangle of \mathbb{R}^2 , we apply a spatial discretization by finite differences scheme which leads to a nonlinear differential system. This system is solved with an implicit Runge-Kutta scheme. At each iteration, we have to solve a nonsymmetric Stein equation. The numerical resolution of the Stein equation is carried out using the global-GMRES method. We then consider the case where the domain Ω is a bounded domain of \mathbb{R}^d , $d \geq 2$. We choose to use a meshless method based on interpolation by radial basis functions. We propose a formalism that differs from the usual approaches found in the litterature. We establish some theoretical results related to the existence of such interpolants using linear algebra techniques. A alternative approach, based on functional analysis is also provided.

The steady and unsteady cases are considered successively. In the first case, the interpolation based on RBF, leads to the resolution of a nonlinear matrix equation $R(X) = 0$. At each iteration of the inexact-Newton algorithm used in this case, we have to solve a linear matrix equation of the form $DR(X) = C$, where $DR(X)$ is the Fréchet derivative of R . As $DR(X)$ cannot be identified to a matrix polynomial of the variable X , we use the global-GMRES algorithm. In the second case, we have to numerically solve a nonlinear differential matrix equation. We adapt the formalism of the implicit Runge-Kutta scheme to the matricial case

and transform the differential problem into a nonlinear matrix equation that is numerically solved as done in the steady case. Eventually, we consider a linear quadratic control problem. We choose the 2D heat equation as a study case. The determination of a control minimizing a given energy leads to the resolution of a Riccati equation. This equation is numerically solved by a ADI preconditioned Newton-Krylov method.

Keywords

Approximation, Arnoldi, Burgers, GMRES, Heat, Krylov, Lyapunov, Meshless, Newton, PDE, RBF, Riccati, Sylvester.

Table des matières

Introduction	13
1 Méthodes de résolution d'équations matricielles linéaires	17
1.1 Introduction	17
1.2 La Méthode GMRES globale	18
1.3 Equation de Sylvester - Lyapunov	22
1.4 Résolution de l'équation de Stein	28
1.5 Exemples numériques	36
1.6 Conclusion	39
2 Résolution d'une équation de type Burgers par différences finies	41
2.1 Introduction	41
2.2 Schéma de résolution par différences finies	43
2.3 Schéma de Runge-Kutta implicite	48
2.4 Méthode quasi-Newton inexacte	49
2.5 Exemples numériques	53
2.6 Conclusion	59
3 Résolution d'une équation de type Burgers par une méthode sans maillage	61
3.1 Introduction	61
3.2 Fonctions à base radiale	62
3.3 Méthodes sans maillage pour une EDP de type Burgers stationnaire	66
3.4 Méthode sans maillage pour une EDP de type Burgers évolutive . .	87
3.5 Conclusion	102
4 Equation matricielle de Riccati continue appliquée au contrôle des équations aux dérivées partielles.	103
4.1 Introduction	103
4.2 Méthode de Newton	105
4.3 Résolution de l'équation de Lyapunov	107

4.4	Algorithme de Newton Arnoldi par blocs	113
4.5	Exemples d'application en contrôle linéaire quadratique	114
4.6	Conclusion	118
	Bibliographie	121

Introduction

Les équations algébriques matricielles sont un outil central pour la résolution des équations aux dérivées partielles et des problèmes de contrôle de systèmes dynamiques issus de leur discrétisation en espace.

Dans le cadre de ce travail, les équations matricielles, issues de la discrétisation en espace d'équations aux dérivées partielles sont, par nature, potentiellement de grande taille. En effet, leur taille est liée au nombre de points de discrétisation. Les méthodes itératives basées sur des projections sur des sous-espaces de Krylov paraissent donc indiquées dans ce cas. Comme nous le verrons dans le cas de la résolution numérique d'EDP non linéaires stationnaires, la discrétisation en espace conduit à un problème de résolution d'une équation matricielle non linéaire. La méthode de Newton est alors appliquée et la construction de la solution approchée passe par la génération des termes d'une suite d'approximations de cette solution. Chacun des termes de la suite en question est obtenu par la résolution d'une équation matricielle linéaire cette fois mais qui présente le désavantage de ne pas toujours s'écrire sous la forme d'une équation polynomiale du premier degré, sauf moyennant peut-être une réécriture des inconnues qui pourrait s'avérer difficile et numériquement coûteuse. L'objectif de ce travail étant aussi d'établir un cadre général en termes de dimension d'espace, nous préférons ne pas réordonner les inconnues de façon à aboutir à un problème de résolution de système linéaire, pour des raisons de taille du système, de difficulté de réécriture du problème et de coût numérique en termes de calculs. Dans ce cas, nous utiliserons l'algorithme GMRES global qui permet une résolution simple. Dans le cas des équations évolutives, on se ramène, par semi-discrétisation (par différences finies ou par une méthode sans maillage) à un système d'équations différentielles ordinaires. Dans les exemples que nous donnons, un schéma d'intégration de Runge-Kutta implicite est utilisé, ce qui donne encore lieu, après linéarisation par la méthode de Newton, au même type d'équations linéaires matricielles à résoudre. Toujours dans le but de résoudre les équations semi discrétisées le plus directement possible, nous les traitons intégralement sous forme matricielle (les inconnues sont donc des matrices de taille $n \times d$, où n est le nombre de points de discrétisation et d est la dimension spatiale du problème. Pour ce faire, le formalisme de la méthode de Runge-Kutta

implicite doit être étendu à ce cadre.

Nous nous intéressons aussi à un cas particulier des équations de Sylvester $AX + XB = C$. Dans certaines situations, notamment tirées de problèmes de contrôle linéaire, les matrices A et B sont creuses, de grande taille et la matrice C peut s'écrire sous la forme d'un produit EF^T où E et F sont de petit rang. Nous proposons une approche, notée LR-BA-ADI (Low Rank Block Arnoldi ADI) consistant à utiliser un préconditionnement de type ADI (Alternating Direction Implicit) qui transforme l'équation de Sylvester en une équation de Stein que nous résolvons par une méthode de Krylov par blocs. Les itérés sont donnés sous forme factorisée, nous permettant d'économiser de la place mémoire. La performance de cette approche au regard d'autres méthodes est confirmée par des tests comparatifs. Enfin, nous proposons une méthode de type Newton-Krylov par blocs avec préconditionnement ADI pour les équations de Riccati issues de problèmes de contrôle linéaire quadratique. Cette méthode est dérivée de la méthode LR-BA-ADI.

Enfin, nous nous penchons sur l'équation matricielle de Riccati $A^T X + XA - XBB^T X + C^T C = 0$ qui intervient notamment dans les problèmes de calcul de contrôle linéaire quadratique. Les méthodes de Krylov existantes offrent en général de bonnes propriétés en terme de rapidité et de vitesse de convergence. Malheureusement, elles ne permettent pas, contrairement aux méthodes basées sur l'algorithme de Newton, la conservation du caractère stabilisant en ce qui concerne la solution approchée. Nous combinons ces deux approches : la méthode de Newton qui à chaque étape nécessitera la résolution d'une équation de Lyapunov -cas particulier de l'équation de Sylvester- qui sera résolue numériquement par la méthode bloc Arnoldi avec préconditionnement de type ADI. Nous veillerons aussi à limiter le temps de calcul et la mémoire utilisée.

Les équations matricielles dont nous discutons les méthodes de résolution dans ce travail se basent sur les situations rencontrées pour la résolution d'EDP. Nous avons choisi l'exemple d'une EDP de type Burgers de la forme $\partial_t u + \mu(u \cdot \nabla)u - \nu Lu = f$ sur un ensemble $\Omega \times [t_0, T]$, où Ω est un domaine de \mathbb{R}^d , $d \geq 2$, μ et ν sont deux paramètres et L est un opérateur différentiel linéaire. Dans le cas où Ω est un rectangle de \mathbb{R}^2 , on applique un schéma de discrétisation en espace par différences finies. On aboutit à un système différentiel non linéaire que nous résolvons par un schéma de Runge-Kutta implicite. Chaque itération donne lieu à la résolution d'une équation matricielle de Stein non symétrique qui sera menée à bien par l'utilisation de la méthode GMRES globale. Dans le cas où le domaine borné Ω est choisi quelconque dans \mathbb{R}^d , $d \geq 2$, les méthodes des éléments finis ou volumes finis sont l'outil standard de discrétisation en espace. Cependant, ils supposent parfois un effort considérable de calcul pour les géométries compliquées. C'est pourquoi nous optons pour une stratégie sans maillage (meshless), basée sur l'interpolation par des fonctions à base radiale (RBF). Les équations matricielles

résultant de cette discrétisation sont encore dans ce cas particulièrement adaptées à la méthode de GMRES globale.

La thèse comporte quatre chapitres :

Dans le premier chapitre, après avoir donné un aperçu des méthodes classiques employées pour les équations de Sylvester et de Lyapunov, on s'intéresse au cas d'équations linéaires générales de la forme $\mathcal{M}(X) = C$, où \mathcal{M} est un opérateur linéaire matriciel. On expose la méthode de GMRES globale qui s'avère particulièrement utile dans le cas où $\mathcal{M}(X)$ ne peut s'exprimer comme un polynôme linéaire en X à coefficients matriciels. On se penche ensuite sur un cas particulier d'équations de Sylvester $AX + XB = C$ pour lequel les matrices A et B sont creuses, de grandes tailles et la matrice C peut s'écrire sous la forme d'un produit EF^T où E et F sont de petit rang. Nous proposons une approche, notée LR-BA-ADI (Low Rank Block Arnoldi ADI) consistant à utiliser un préconditionnement de type ADI qui transforme l'équation de Sylvester en une équation de Stein que nous résolvons par une méthode de Krylov par blocs. Les itérés sont donnés sous forme factorisée, nous permettant ainsi d'économiser de la place mémoire. La performance de cette approche au regard d'autres méthodes est confirmée par des tests comparatifs. Des résultats de convergence et de majoration de l'erreur sont donnés.

Dans le deuxième chapitre, nous nous penchons sur une équation de type Burgers en deux dimensions sur un rectangle. Par semi-discrétisation par la méthode des différences finies, nous obtenons un système d'équations différentielles ordinaires. Le schéma d'intégration de Runge-Kutta implicite que nous utilisons dans ce cas donne lieu à la résolution d'une équation vectorielle non linéaire. En lui appliquant l'algorithme de Newton, nous devons résoudre des systèmes linéaires que nous transformons en équations de Stein, résolues par la méthode du GMRES globale. Des test numériques sont donnés pour illustrer notre approche.

Dans troisième chapitre, nous généralisons le problème de la résolution de l'équation de type Burgers au cas où le domaine est quelconque. Après avoir rappelé des généralités sur l'interpolation par des fonctions à base radiale, nous proposons une méthode sans maillage pour la résolution numérique d'une équation de type Burgers. Nous proposons un formalisme différent de ce qui est habituellement donné dans la littérature. Nous énonçons les résultats théoriques garantissant l'existence de tels interpolants en faisant appel à des techniques d'algèbre linéaire. Les cas stationnaire et évolutif sont traités successivement. Dans le premier cas, par interpolation par des fonctions à base radiale, on se ramène à la résolution d'une équation matricielle non linéaire $R(X) = 0$. La méthode de Newton-inexacte que nous employons demande, à chaque pas, la résolution d'une équation linéaire ma-

tricielle de la forme $DR(X) = C$, où $DR(X)$ est la dérivée de Fréchet de R , qui ne peut être identifiée à un polynôme en X à coefficients matriciels, nous amenant alors à utiliser la méthode GMRES globale. Dans le second cas, on se ramène à la résolution d'une équation différentielle ordinaire matricielle non linéaire. On adapte le formalisme de la méthode de Runge-Kutta implicite au cas matriciel et comme dans le cas stationnaire, on se ramène à un problème de résolution d'une équation matricielle non linéaire qui sera traitée de la même façon. Des tests numériques sont donnés pour illustrer notre approche.

Le dernier chapitre porte sur les équations de Riccati liées aux problèmes de contrôle linéaire. Nous nous basons sur deux exemples liés à l'équation de la chaleur. Nous mettons en oeuvre une méthode basée sur l'algorithme de Newton pour la résolution de l'équation de Riccati. Après avoir énoncé les conditions de convergence de la méthode et surtout le caractère stabilisant de la solution approchée, chaque itération de l'algorithme de Newton requiert la résolution d'une équation de Lyapunov. En utilisant un préconditionnement de type ADI, nous résolvons ces équations de Lyapunov par l'algorithme d'Arnoldi par blocs, méthode que nous appelons Newton-Block Arnoldi-ADI. Des essais numériques sont proposés et nous comparons les résultats obtenus par notre méthode à ceux produits par l'algorithme de LRCF-Newton-ADI.

Méthodes de résolution d'équations matricielles linéaires

1.1 Introduction

Les équations matricielles interviennent dans de nombreux domaines des mathématiques, de la mécanique, des sciences physiques et de l'automatisme. On peut citer par exemple la recherche de sous-espaces invariants en géométrie ou en algèbre linéaire, les problèmes de valeurs propres, la résolution d'équations différentielles ordinaires ou aux dérivées partielles, le contrôle linéaire quadratique et la réduction de modèle.

Dans ce chapitre, nous nous intéressons à quelques méthodes de résolution d'équations matricielles linéaires. Pour les problèmes de taille réduite, des méthodes directes sont bien connues et sont intégrées dans les logiciels de calcul numérique comme Matlab ou Scilab. Ces méthodes sont basées sur des décompositions de matrices (citons, en ce qui concerne la résolution des équations de Lyapunov, Sylvester et Stein, la méthode de Bartel-Stewart, basée sur la décomposition de Schur des matrices [5], la méthode de Hammarling utilisant la décomposition de Cholesky) [42, 43]. Ces méthodes 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 $\mathcal{O}(10^5)$, le temps CPU ainsi que la taille mémoire peuvent poser problème. La méthode itérative dite du matrix sign function peut-être utilisée pour des problèmes de taille moyenne [30]. Les méthodes que nous venons d'évoquer ne peuvent cependant pas être utilisées pour des problèmes de grande taille. On utilise alors des méthodes itératives consistant en la projection du problème initial sur une suite croissante de sous-espaces de Krylov, transformant le problème en un problème de taille réduite qui peut être résolu par des méthodes directes et dont la solution est une approximation de la solution du problème initial. Parmi les nombreux ouvrages de référence traitant des méthodes de Krylov et de l'algorithme GMRES, on pourra citer [67, 77]. Ces méthodes diffèrent l'une de l'autre par le type de projection qui est appliquée et le choix des sous-espaces de Krylov.

On peut citer les méthodes GMRES par bloc [78, 73, 74, 79, 34], GMRES étendu [80, 44], GMRES rationnel [6] par exemple.

Ce chapitre comporte deux parties : la première est consacrée aux équations matricielles linéaires dans un cas général. Pour ces équations, qui interviendront dans les chapitres 2 et 3 de ce travail, nous appliquerons la méthode GMRES globale [53].

La deuxième partie est consacrée à l'équation de Sylvester avec un second membre de petit rang. Nous proposerons une nouvelle méthode, basée sur l'algorithme d'Arnoldi par blocs associé à un préconditionnement de type ADI. Nous donnerons des résultats de convergence et de majoration de l'erreur et illustrerons ses performances par des tests numériques en la comparant avec les méthodes les plus utilisées actuellement.

1.2 La Méthode GMRES globale

On considère l'équation matricielle linéaire

$$\mathcal{M}(\Lambda) = C, \tag{1.1}$$

où $\mathcal{M} : \mathbb{R}^{n \times d} \rightarrow \mathbb{R}^{n \times d}$ est un opérateur linéaire matriciel bijectif, l'inconnue Λ est une matrice réelle de taille $n \times d$ et C est une matrice de taille $n \times d$.

Soit V une matrice réelle de taille $n \times d$. Soit i un entier naturel, on définit la matrice $\mathcal{M}^i(V)$ de $\mathbb{R}^{n \times d}$ par la relation de récurrence $\mathcal{M}^i(V) = \mathcal{M}(\mathcal{M}^{i-1}(V))$, avec la convention $\mathcal{M}^0(V) = V$. On définit le sous-espace vectoriel, appelé espace de Krylov, $\mathcal{K}_l(\mathcal{M}, V) = \text{sev}\{V, \mathcal{M}(V), \dots, \mathcal{M}^{l-1}(V)\}$, engendré par les matrices $V, \mathcal{M}(V), \dots, \mathcal{M}^{l-1}(V)$, où l est un entier naturel non nul. L'ensemble $\mathcal{K}_l(\mathcal{M}, V)$ ainsi défini est un sous-espace vectoriel de $\mathbb{R}^{n \times d}$.

Soient $M \in \mathbb{R}^{p \times q}$ et $N \in \mathbb{R}^{r \times s}$ deux matrices. On définit le produit de Kronecker $M \otimes N$ de M par N comme étant la matrice par blocs

$$M \otimes N = \begin{pmatrix} m_{1,1}N & \dots & m_{1,q}N \\ \vdots & & \vdots \\ m_{p,1}N & \dots & m_{p,q}N \end{pmatrix} \in \mathbb{R}^{pr \times qs}. \tag{1.2}$$

On définit l'opération *vec* qui à une matrice associe le vecteur constitué par ses colonnes empilées les unes sur les autres dans l'ordre

$$\text{vec}(M) = [m_{1,1}, \dots, m_{p,1}, \dots, m_{1,q}, \dots, m_{p,q}]^T \in \mathbb{R}^{pq}. \tag{1.3}$$

Nous donnons sans les démontrer les propriétés suivantes qui nous seront utiles dans la suite de notre exposé.

Proposition 1.1. *Soient M , N , P et Q quatre matrices. Sous réserve de compatibilité des tailles des facteurs, on a l'identité*

$$(M \otimes N)(P \otimes Q) = (MP) \otimes (NQ). \quad (1.4)$$

Proposition 1.2. *Soient $U \in \mathbb{R}^{p \times p}$ et $V \in \mathbb{R}^{r \times r}$ deux matrices carrées. Le spectre $\Lambda(U \otimes V)$ de la matrice $U \otimes V \in \mathbb{R}^{pr \times pr}$ est donné par*

$$\Lambda(U \otimes V) = \{\lambda_i \cdot \mu_j\}_{1 \leq i \leq p, 1 \leq j \leq r}.$$

où

$$\{\lambda_1, \dots, \lambda_p\} = \Lambda(U) \text{ et } \{\mu_1, \dots, \mu_r\} = \Lambda(V)$$

désignent les spectres de U et de V respectivement.

Proposition 1.3. *Soient $M \in \mathbb{R}^{p \times q}$ et $N \in \mathbb{R}^{r \times s}$ deux matrices. Pour toute matrice X de $\mathbb{R}^{q \times r}$, nous avons l'identité*

$$\text{vec}(MXN) = (N^T \otimes M)\text{vec}(X). \quad (1.5)$$

Un exposé exhaustif et en particulier les démonstrations des propositions 1.1, 1.2 et 1.3 peuvent être trouvés dans [47, 62, 64].

On rappelle la définition du produit scalaire de Frobenius de deux matrices A et B de taille $n \times d$ par $\langle A, B \rangle_F = \text{tr}(A^T B)$ où $\text{tr}(A^T B)$ désigne la trace de la matrice carrée $A^T B$. La norme induite est définie par $\|A\|_F = \sqrt{\langle A, A \rangle_F}$.

Soit V une matrice réelle de taille $n \times d$. L'algorithme d'Arnoldi global modifié, dont les étapes sont données ci-dessous, produit une base orthonormale $\{V_1, \dots, V_l\}$ de l'espace de Krylov $\mathcal{K}_l(\mathcal{M}, V)$ au sens de la norme de Frobenius, c'est à dire telle que $\langle V_i, V_j \rangle_F = \delta_{i,j}$, $1 \leq i, j \leq l$.

Algorithm 1 L'algorithme d'Arnoldi global modifié

- Initialisation : $V_1 = V/\|V\|_F$
 - **Pour** $j = 1 : l$
 - Calculer $\tilde{V}_j = \mathcal{M}(V_j)$
 - **Pour** $i = 1 : j$
 - Calculer $h_{i,j} = \langle V_i, \tilde{V}_j \rangle_F$
 - $\tilde{V}_j = \tilde{V}_j - h_{i,j}V_i$
 - **Fin Pour** i
 - Calculer $h_{j+1,j} = \|\tilde{V}_j\|$
 - Calculer $V_{j+1} = \tilde{V}_j/h_{j+1,j}$
 - **Fin pour** j
-

On définit la matrice $\mathcal{V}_l = [V_1, \dots, V_l] \in \mathbb{R}^{n \times dl}$ et \tilde{H}_l , la matrice de Hessenberg supérieure de taille $(l+1) \times l$ dont les entrées non nulles sont calculées par l'algorithme d'Arnoldi.

$$\tilde{H}_l = \begin{bmatrix} h_{11} & \dots & \dots & \dots & h_{1,l} \\ h_{21} & \ddots & & & \vdots \\ 0 & \ddots & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \ddots & \vdots & \vdots & h_{l,l} \\ 0 & \dots & \dots & 0 & h_{l+1,l} \end{bmatrix} \quad (1.6)$$

La matrice H_l de taille $l \times l$ est obtenue en supprimant la dernière ligne de \tilde{H}_l . Par construction, la matrice par blocs \mathcal{V}_l est F-orthonormale, ce qui signifie que les matrices V_1, \dots, V_l constituent un système orthonormal au sens du produit scalaire de Frobenius. On établit les identités suivantes de manière immédiate

$$[\mathcal{M}(V_1), \dots, \mathcal{M}(V_l)] = \mathcal{V}_l(H_l \otimes I_d) + E_{l+1}, \quad (1.7)$$

où

$$E_{l+1} = h_{l+1,l}[0_{n \times d}, \dots, 0_{n \times d}, V_{l+1}],$$

et

$$[\mathcal{M}(V_1), \dots, \mathcal{M}(V_l)] = \mathcal{V}_{l+1}(\tilde{H}_l \otimes I_d). \quad (1.8)$$

La méthode GMRES globale consiste à construire itérativement une suite $(\Lambda_l)_{l \in \mathbb{N}^*}$ d'approximations de la solution Λ^* de l'équation (1.1) de la façon suivante. On commence par choisir un premier terme $\Lambda_0 \in \mathbb{R}^{n \times d}$ et en notant \mathcal{R}_0 le résidu correspondant : $\mathcal{R}_0 = C - \mathcal{M}(\Lambda_0)$, on construit les itérés Λ_l de telle manière à avoir

$$\Lambda_l = \Lambda_0 + Z_l \text{ où } Z_l \in \mathcal{K}_l(\mathcal{M}, \mathcal{R}_0) \quad (1.9)$$

$$\mathcal{R}_l = C - \mathcal{M}(\Lambda_l) \perp_F \mathcal{K}_l(\mathcal{M}, \mathcal{M}(\mathcal{R}_0)). \quad (1.10)$$

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

$$\min_{\Lambda \in \Lambda_0 + \mathcal{K}_l(\mathcal{M}, \mathcal{R}_0)} \|C - \mathcal{M}(\Lambda)\|_F \quad (1.11)$$

Le résultat suivant [18] établit que le problème de moindres carrés (1.11) est équivalent à un problème de dimension réduite

Proposition 1.4. *À l'étape l , l'approximation Λ_l construite par la méthode GMRES globale est donnée par*

$$\Lambda_l = \Lambda_0 + \mathcal{V}_l(y_l \otimes I_d),$$

où y_l est solution du problème aux moindres carrés

$$\min_{y \in \mathbb{R}^l} \|\|\mathcal{R}_0\|_F e_1 - \tilde{H}_l y\|_2 \quad (1.12)$$

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

Pour résoudre le problème (1.12), considérons la décomposition QR de la matrice \tilde{H}_l

$$\tilde{H}_l : \tilde{R}_l = Q_l \tilde{H}_l,$$

où $\tilde{R}_l \in \mathbb{R}^{(l+1) \times l}$ est triangulaire supérieure et $Q_l \in \mathbb{R}^{(l+1) \times (l+1)}$ est orthogonale. Posons $g_l = \|\mathcal{R}_0\|_F Q_l e_1$. En notant R_l la matrice $l \times l$ obtenue en éliminant la dernière ligne de \tilde{R}_l , le vecteur y_l est calculé en résolvant le système triangulaire $R_l y_l = g_l$.

À chaque itération, le résidu \mathcal{R}_l doit être calculé, ce qui peut s'avérer coûteux. Le résultat donné par la proposition suivante y remédie (on pourra se référer à [18] pour les détails) en établissant que $\|\mathcal{R}_l\|_F$ peut être calculée sans évaluer $\mathcal{M}(\Lambda_l)$.

Proposition 1.5. *À l'étape l , le résidu $\mathcal{R}_l = C - \mathcal{M}(\Lambda_l)$ obtenu par la méthode GMRES globale vérifie les deux identités suivantes*

$$\mathcal{R}_l = \gamma_{l+1} \mathcal{V}_{l+1}(Q^T e_{l+1} \otimes I_d) \quad (1.13)$$

et

$$\|\mathcal{R}_l\|_F = |\gamma_{l+1}|, \quad (1.14)$$

où γ_{l+1} est la dernière composante du vecteur $g_l = \|\mathcal{R}_0\|_F Q_l e_1$ et e_{l+1} est le dernier vecteur de la base canonique de $\mathbb{R}^{l+1} : e_{l+1} = (0, \dots, 0, 1)^T$.

À chaque nouvelle itération de l'algorithme GMRES globale, la taille des matrices produites par la méthode d'Arnoldi augmente entraînant des coûts en termes de temps de calcul et de mémoire de plus en plus importants. Pour y remédier, on

adopte une stratégie de redémarrage après un nombre choisi d'itérations. A chaque redémarrage, on choisit la dernière approximation calculée comme terme initial pour l'algorithme de GMRES global, résumé ci-après.

Algorithm 2 Algorithme GMRES global avec redémarrage

1. Initialisation : on choisit Λ_0 , une tolérance ϵ et on pose $iter = 0$
 2. Calculer $\mathcal{R}_0 = C - \mathcal{M}(\Lambda_0)$, $\beta = \|\mathcal{R}_0\|$, et $V_1 = \mathcal{R}_0/\beta$
 3. Executer l'algorithme 1 pour construire une base F-orthonormale $\{V_1, \dots, V_l\}$ de $\mathcal{K}_l(\mathcal{M}, V)$
 4. Calculer y_l réalisant $\min_{y \in \mathbb{R}^l} \|\|\mathcal{R}_0\|_{Fe_1} - \tilde{H}_l y\|_2$
 5. Calculer $\Lambda_l = \Lambda_0 + \mathcal{V}_l(y_l \otimes I_d)$
 6. Calculer la norme du résidu $\|\mathcal{R}_l\|_F$ en utilisant la proposition (1.14)
 7. **Si** $\|\mathcal{R}_l\|_F < \epsilon$
 8. **Arrêt**
 9. **Sinon**
 10. $\Lambda_0 = \Lambda_l$, $\mathcal{R}_0 = \mathcal{R}_l$, $\beta = \|\mathcal{R}_0\|_F$, $V_1 = \mathcal{R}_0/\beta$, $iter=iter+1$, **Aller à 2** :
 11. **Fin si**
-

Dans les chapitres suivants, nous aurons l'occasion d'appliquer cette méthode de résolution à des équations linéaires matricielles qui se présentent notamment sous la forme générale $\mathcal{M}(X) = C$, où \mathcal{M} désigne un endomorphisme d'un espace de matrices.

1.3 Equation de Sylvester - Lyapunov

1.3.1 Considérations générales

Dans cette partie, nous considérons une équation matricielle de la forme

$$AX + XB = C \tag{1.15}$$

où $A \in \mathbb{R}^{m \times m}$, $B \in \mathbb{R}^{n \times n}$, C et $X \in \mathbb{R}^{m \times n}$.

L'importance que revêt cette équation réside dans la multiplicité des domaines où elle intervient : calcul de sous-espaces invariants, réduction de modèle en théorie du contrôle (calcul de grammien), calcul d'un contrôle optimal en contrôle linéaire quadratique, résolution de l'équation de Riccati, discrétisation d'équations aux dérivées partielles.

Intéressons-nous à l'existence d'une solution de l'équation 1.15. On démontre facilement que l'équation 1.15 peut s'écrire comme l'équation vectorielle

$$(I_n \otimes A + B^T \otimes I_m) \text{vec}(X) = \text{vec}(C), \quad (1.16)$$

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

Cette écriture qui ramène le problème à la résolution d'un système linéaire, ne présente en réalité que peu d'avantage en terme de résolution, qu'elle soit directe ou itérative : dans bien des situations, la taille du système à résoudre est très importante. En revanche, cette formulation permet d'établir, à la lumière de la proposition 1.3, une condition nécessaire et suffisante d'existence et d'unicité de la solution de l'équation 1.15.

Proposition 1.6. *L'équation $AX + XB = C$ admet une solution unique si et seulement si les spectres des matrices A et $-B$ sont disjoints.*

Cette proposition établit l'existence et l'unicité de la solution de l'équation 1.15 sans en donner une écriture explicite. Des hypothèses plus fortes nous permettent de donner une écriture de la solution.

Définition 1.7. Une matrice carrée est dite stable si toutes ses valeurs propres appartiennent au sous ensemble $\mathbb{R}_-^* + i\mathbb{R}$ de \mathbb{C} .

Proposition 1.8. *Si les matrices A et B sont stables, alors l'équation 1.15 admet pour unique solution la matrice*

$$X = - \int_0^{+\infty} e^{tA} C e^{tB} dt.$$

La méthode de Hessenberg-Schur [39] qui est une modification de l'algorithme de Bartels-Stewart [5] est devenue standard pour la résolution directe des équations de Sylvester de taille petite à modérée. 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. Cependant, ces méthodes ne sont guère utilisables lorsque A ou B est grande ou creuse.

Dans le cas où les matrices A , B sont de taille moyenne ou grande, on est amené à utiliser des méthodes itératives. Les nombreuses méthodes de projections sur des sous-espaces de Krylov, basées sur un algorithme de type Arnoldi [48, 34, 52, 76, 80, 49, 72] sont intéressantes si les matrices A , B sont creuses et lorsqu'on ne dispose pas d'informations sur leur spectre. Il existe d'autres méthodes itératives telles que la méthode de Newton ou la méthode de la fonction signe de matrice qui sont adaptées au cas où les matrices A et B sont denses [75]. Les méthodes de Smith [82, 71] et la méthode ADI (alternating directional implicit) [35, 7, 27, 48] peuvent aussi être employées si l'on connaît les spectres des matrices

A et B . La méthode ADI fut introduite par Peaceman et Rachford en 1955 [69] pour la résolution de systèmes linéaires tirés de la discrétisation d'équations aux dérivées partielles de type elliptique. Plus tard, cette méthode a été adaptée pour la résolution d'équations matricielles de type Lyapunov et Sylvester [82, 71, 65]. Elle repose sur le schéma suivant :

- On choisit un premier terme $X_0 \in \mathbb{R}^{m \times n}$ et une suite de paramètres complexes $\{\lambda_i\}$ et $\{\mu_i\}$
- On calcule la suite des approximations X_i de la solution exacte X en résolvant le système

$$\text{Calculer } X_{i+\frac{1}{2}} \quad \text{tel que } (A + \mu_i I_n) X_{i+\frac{1}{2}} = C - X_i (B - \mu_i I_s), \quad (1.17)$$

$$\text{Calculer } X_{i+1} \quad \text{tel que } X_{i+1} (B + \lambda_i I_s) = C - (A - \lambda_i I_n) X_{i+\frac{1}{2}}, \quad (1.18)$$

En particulier, la méthode ADI permet d'obtenir une convergence plus rapide que les méthodes de Krylov sous réserve que les paramètres λ_i et μ_i soient choisis de telle manière que les systèmes (1.17) et (1.18) puissent être résolus rapidement. Nous allons à présent nous pencher sur le cas où l'équation de Sylvester présente la particularité suivante : les matrices A et B sont creuses et le second membre C peut s'écrire comme le produit EF^T de deux matrices, où $E \in \mathbb{R}^{m \times r}$ et $F \in \mathbb{R}^{n \times r}$ sont de rang maximal $r \ll m, n$. Cette situation se présente en effet dans certains problèmes liés notamment à la théorie du contrôle linéaire, et nous allons montrer comment tirer parti des avantages respectifs de la méthode ADI et des méthodes de Krylov en exploitant la structure particulière des matrices de coefficients A , B , E et F .

1.3.2 La Méthode de type Arnoldi par blocs avec préconditionnement ADI

Nous exposons dans cette partie une méthode de type Arnoldi par blocs avec préconditionnement de type ADI dans le cas où l'équation 1.15 s'écrit

$$AX + XB = EF^T \quad (1.19)$$

où les matrices $A \in \mathbb{R}^{m \times m}$, $B \in \mathbb{R}^{n \times n}$ sont creuses et de grande taille et le second membre s'écrit comme le produit de deux matrices $E \in \mathbb{R}^{m \times r}$ et $F \in \mathbb{R}^{n \times r}$ de rang maximal r , avec $r \ll m, n$. On rencontre cette situation notamment dans la résolution numérique d'équation différentielles matricielles de Riccati, en restauration d'image. Pour de plus amples détails, on pourra se référer à [35, 27, 48] et [14, 29, 7].

Nous supposons que les matrices A et B sont stables afin de s'assurer de l'existence et de l'unicité d'une solution de l'équation (1.19). Nous allons décrire une approche

permettant d'accélérer la convergence de la méthode de Krylov par blocs [34, 52] en la combinant à un préconditionnement de type ADI [65, 71].

Soient $\{\lambda_1, \lambda_2, \dots\}$ et $\{\mu_1, \mu_2, \dots\}$ deux ensembles de paramètres complexes et $X_0 \in \mathbb{R}^{m \times n}$ une matrice choisie comme première approximation de la solution X de (1.19). La méthode ADI repose sur le schéma suivant qui consiste en l'alternance des deux systèmes linéaires matriciels (1.17) et (1.18), pour $i = 0, 1, \dots$

Les scalaires λ_i et μ_i sont des paramètres complexes ou réels choisis de telle sorte que la suite des approximations successives X_i converge rapidement vers la solution exacte X de l'équation de Sylvester (1.19). En combinant les équations (1.17) et (1.18), nous obtenons une expression de X_{i+1} en fonction de X_i

$$\begin{aligned} X_{i+1} &= (A - \lambda_i I_n) (A + \mu_i I_n)^{-1} X_i (B - \mu_i I_s) (B + \lambda_i I_s)^{-1} \\ &\quad + (\lambda_i + \mu_i) (A + \mu_i I_n)^{-1} E F^T (B + \lambda_i I_s)^{-1}. \end{aligned} \quad (1.20)$$

En remarquant que X est solution des équations (1.17) et (1.18), on a aussi

$$\begin{aligned} X &= (A - \lambda_i I_n) (A + \mu_i I_n)^{-1} X (B - \mu_i I_s) (B + \lambda_i I_s)^{-1} \\ &\quad + (\lambda_i + \mu_i) (A + \mu_i I_n)^{-1} E F^T (B + \lambda_i I_s)^{-1}. \end{aligned} \quad (1.21)$$

L'erreur $E_i := X_i - X$ (pour $i = 1, 2, \dots$) à l'étape i est obtenue par soustraction membre à membre de (1.21) et de (1.20)

$$\begin{aligned} E_{i+1} &= (A - \lambda_i I_n) (A + \mu_i I_n)^{-1} E_i (B - \mu_i I_s) (B + \mu_i I_s)^{-1} \\ &= \left(\prod_{j=0}^i (A - \lambda_j I_n) (A + \mu_j I_n)^{-1} \right) E_0 \left(\prod_{j=0}^i (B - \mu_j I_s) (B + \lambda_j I_s)^{-1} \right) \end{aligned} \quad (1.22)$$

Remarquons que si le spectre de A est inclus dans l'ensemble des paramètres $\{\lambda_1, \lambda_2, \dots\}$ et/ou le spectre de B est inclus dans $\{\mu_1, \mu_2, \dots\}$, alors par le théorème de Cayley-Hamilton, on a $E_{i+1} = 0$ et donc $X = X_{i+1}$.

La question du choix des paramètres se pose dans l'objectif d'obtenir la convergence la plus rapide possible. La relation (1.22) implique la majoration

$$\begin{aligned} \frac{\|E_{i+1}\|}{\|E_0\|} &\leq \prod_{j=0}^i \|(A - \lambda_j I) (A + \mu_j I)^{-1}\| \|(B - \mu_j I) (B + \lambda_j I)^{-1}\|, \\ &\leq \max_{\alpha \in \sigma(A), \beta \in \sigma(B)} \prod_{j=0}^i \left| \frac{(\alpha - \lambda_j) (\beta - \mu_j)}{(\alpha + \mu_j) (\beta + \lambda_j)} \right| \end{aligned} \quad (1.23)$$

Cette dernière inégalité nous amène à rechercher les paramètres réalisant le problème de minimisation

$$\{\lambda_1, \dots, \lambda_i, \mu_1, \dots, \mu_i\} = \arg \min \left(\max_{\alpha \in \Lambda(A), \beta \in \Lambda(B)} \prod_{j=0}^i \left| \frac{(\alpha - \lambda_j)(\beta - \mu_j)}{(\alpha + \mu_j)(\beta + \lambda_j)} \right| \right). \quad (1.24)$$

Remarquons que l'inégalité (1.23) implique

$$\frac{\|E_{i+1}\|}{\|E_0\|} \leq \max_{\alpha \in \Lambda(A)} \prod_{j=0}^i \left| \frac{(\alpha - \lambda_j)}{(\alpha + \mu_j)} \right| \cdot \max_{\beta \in \Lambda(B)} \prod_{j=0}^i \left| \frac{(\beta - \mu_j)}{(\beta + \lambda_j)} \right|, \quad (1.25)$$

ce qui suggère de rechercher des paramètres pseudo-optimaux réalisant

$$\{\lambda_1, \dots, \lambda_i, \mu_1, \dots, \mu_i\} = \arg \min \left(\max_{\alpha \in \Lambda(A)} \prod_{j=0}^i \left| \frac{(\alpha - \lambda_j)}{(\alpha + \mu_j)} \right| \cdot \max_{\beta \in \Lambda(B)} \prod_{j=0}^i \left| \frac{(\beta - \mu_j)}{(\beta + \lambda_j)} \right| \right). \quad (1.26)$$

Les problèmes de minimax énoncés précédemment sont encore des questions non résolues dans le cas général. Pour l'équation de Lyapunov (c'est à dire lorsque $B = A^T$), Penzl [71] donne un procédé économique pour calculer des paramètres sous-optimaux en utilisant l'algorithme d'Arnoldi classique. Pour l'équation de Sylvester, on peut se référer aux travaux de Wachspress, notamment [85]. La technique de Penzl a été étendue dans [11] pour l'équation de Sylvester. La stratégie qui sera détaillée dans la section suivante, repose sur la résolution du problème de minimax du type

$$\{\mu_1, \mu_2, \dots, \mu_l\} = \arg \min_{(\mu_1, \mu_2, \dots, \mu_l) \in \mathbb{C}_-^l} \left(\max_{\lambda \in \sigma(A)} \frac{|(\lambda - \mu_1) \dots (\lambda - \mu_l)|}{|(\lambda + \mu_1) \dots (\lambda + \mu_l)|} \right),$$

non plus sur les spectres de A et B - qui ne sont généralement pas connus - mais sur des ensembles discrets constitués de valeurs de Ritz approximant les valeurs extrémales des spectres. Le calcul des paramètres par la procédure mise au point par Penzl peut ne pas fonctionner dans le cas où les valeurs de Ritz ne permettent pas d'obtenir une assez bonne estimation des valeurs propres. Il est à noter que dans le cas de l'équation de Lyapunov, la fonction `lp-para` de Matlab (R) de la bibliothèque `LYAPACK` [70] permet de calculer ces paramètres sous optimaux. Nous allons voir comment préconditionner l'équation de Sylvester (1.19) en utilisant deux paramètres ADI sous optimaux et en écrivant les approximations successives sous la forme de produits de deux facteurs de petit rang. Cette méthode, que

nous noterons LR-ADI(2) (Low-Rank-ADI(2)), où le 2 fait référence au nombre de paramètres ADI choisis, est l'objet du paragraphe suivant.

1.3.3 La méthode d'Arnoldi par blocs préconditionnée LR-ADI(2).

La méthode que nous exposons dans cette section repose sur le préconditionnement de l'équation de Sylvester (1.19) par le choix de deux paramètres ADI λ et μ réels strictement négatifs. D'après la relation (1.21), l'équation (1.19) est équivalente à l'équation de Stein non symétrique (appelée aussi équation de Sylvester discrète)

$$\mathcal{A}_\mu X \mathcal{B}_\mu - X = \mathcal{E}_\mu \mathcal{F}_\mu^T, \quad (1.27)$$

où

$$\mathcal{A}_{\lambda,\mu} = (A - \lambda I_m)(A + \mu I_m)^{-1}, \quad \mathcal{B}_{\lambda,\mu} = (B - \mu I_n)(B + \lambda I_n)^{-1}, \quad (1.28)$$

$$\mathcal{E}_{\lambda,\mu} = \sqrt{-(\lambda + \mu)}(A + \mu I_m)^{-1}E \quad \text{et} \quad \mathcal{F}_{\lambda,\mu} = \sqrt{-(\lambda + \mu)}(B + \lambda I_n)^{-1}F. \quad (1.29)$$

Remarquons que la stabilité des matrices A et B assure l'existence des matrices $\mathcal{A}_{\lambda,\mu}$ et $\mathcal{B}_{\lambda,\mu}$.

Les considérations faites sur le choix des paramètres ADI nous suggèrent de rechercher des paramètres λ et μ tels que l'on ait

$$\{\lambda, \mu\} = \operatorname{argmin} \left(\max_{\alpha \in \Lambda(A)} \left| \frac{\alpha - \lambda}{\alpha + \mu} \right| \cdot \max_{\beta \in \Lambda(B)} \left| \frac{\beta - \mu}{\beta + \lambda} \right| \right). \quad (1.30)$$

Nous proposons une adaptation de la routine lp-para de la bibliothèque LYAPACK [11, 70] au cas de l'équation de Sylvester. Les étapes de cet algorithme sont détaillées ci-après.

Algorithm 3 Calcul des paramètres LR-ADI(2)

-
- Initialisation : on choisit deux vecteurs initiaux $r_1 \in \mathbb{R}^m \setminus \{0\}$ et $r_2 \in \mathbb{R}^n \setminus \{0\}$.
 - Exécuter k_+ itérations de l'algorithme d'Arnoldi sur (A, r_1) et sur (B, r_2) pour générer deux ensembles \mathcal{R}_A^+ et \mathcal{R}_B^+ de valeurs de Ritz approximant les plus grandes valeurs propres de A et de B .
 - Exécuter k_- itérations de l'algorithme d'Arnoldi sur (A^{-1}, r_1) et sur (B^{-1}, r_2) pour générer deux ensembles \mathcal{R}_A^- et \mathcal{R}_B^- de valeurs de Ritz approximant les inverses des plus petites valeurs propres de A et de B .
 - On définit les ensembles $\mathcal{R}_A = \mathcal{R}_A^+ \cup 1/\mathcal{R}_A^- = [\rho_1^A, \dots, \rho_k^A]$ et $\mathcal{R}_B = \mathcal{R}_B^+ \cup 1/\mathcal{R}_B^- = [\rho_1^B, \dots, \rho_k^B]$, où $k = k_+ + k_-$.
 - **pour** $i = 1$ **à** k
 - **pour** $j = 1$ **à** k
 - Déterminer $x(i, j) = \max_{x \in \mathcal{R}_A} \left| \frac{x - \rho_i^A}{x + \rho_j^B} \right|$
 - Déterminer $y(i, j) = \max_{x \in \mathcal{R}_B} \left| \frac{x - \rho_i^B}{x + \rho_j^A} \right|$
 - On pose $w(i, j) = x(i, j) \cdot y(i, j)$.
 - **fin pour** j
 - **fin pour** i
 - Détecter $w(i_0, j_0) = \min_{i,j} |w(i, j)|$.
 - Choix des paramètres : $\lambda = \rho_{i_0}^A$ et $\mu = \rho_{j_0}^B$.
-

Le choix de grandes valeurs de k_+ et k_- permet d'avoir de meilleures approximations des spectres de A et de B mais augmente le temps de calcul qui nécessite k_+ produits matrice-vecteur avec A et B et k_- systèmes linéaires à résoudre avec A et B . Dans nos tests numériques, nous avons choisi $k_+ = k_- = 20$ et avons utilisé la sous-routine lp-para de LYAPACK [70] pour le calcul de ces paramètres sous-optimaux.

Une fois les paramètres choisis, on s'intéresse à la résolution de l'équation de Stein (1.27) obtenue après application du préconditionnement à l'équation (1.19).

1.4 Résolution de l'équation de Stein

Dans ce paragraphe, nous considérons l'équation de Stein non symétrique (1.27) dans laquelle nous omettons de faire mention des indices dans le souci d'alléger les notations

$$\mathcal{A} X \mathcal{B} - X = \mathcal{E} \mathcal{F}^T, \quad (1.31)$$

où $\mathcal{A} \in \mathbb{R}^{m \times m}$, $\mathcal{B} \in \mathbb{R}^{n \times n}$, $\mathcal{E} \in \mathbb{R}^{m \times r}$ et $\mathcal{F}^{n \times r}$ sont les matrices explicitées dans (1.28) et (1.29).

La proposition suivante [62] précise les conditions d'existence d'une solution unique.

Proposition 1.9. *L'équation (1.31) admet solution unique si et seulement si $\lambda_i \cdot \mu_j \neq 1$, pour tout $\lambda_i \in \Lambda(\mathcal{A})$ et tout $\mu_j \in \Lambda(\mathcal{B})$, $i = 1, \dots, m$, $j = 1, \dots, n$.*

En renforçant les hypothèses, on a

Proposition 1.10. [62] *Si $\Lambda(\mathcal{A})$ et $\Lambda(\mathcal{B})$ sont inclus dans le disque unité ouvert de \mathbb{C} , alors, l'équation (1.31) admet une solution unique X qui peut s'exprimer sous la forme*

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

Nous nous référons à [62] 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 cet exposé. Dans la section suivante, nous explicitons la résolution numérique de l'équation (1.31) par la méthode d'Arnoldi par blocs.

1.4.1 La méthode d'Arnoldi par blocs pour l'équation de Stein

Les matrices \mathcal{E} et \mathcal{F} étant supposées avoir un rang r petit par rapport aux tailles de \mathcal{A} et \mathcal{B} , nous allons chercher à construire une suite d'approximations de petit rang $(X_l)_{l \in \mathbb{N}^*}$ de la solution X sous la forme

$$X_l = \mathcal{V}_l Y_l \mathcal{W}_l^T, \quad (1.32)$$

où $\mathcal{V}_l, \mathcal{W}_l$ sont les matrices orthonormales construites en appliquant simultanément l itérations de l'algorithme d'Arnoldi par blocs [49, 50] aux paires de matrices $(\mathcal{A}, \mathcal{E})$ et $(\mathcal{B}^T, \mathcal{F})$ respectivement. Dans l'algorithme suivant, nous détaillons les étapes de cet algorithme appliqué à la paire $(\mathcal{A}, \mathcal{E})$.

1.4.2 L'algorithme d'Arnoldi par blocs

Soient $\mathbb{K}_l(\mathcal{A}, \mathcal{E}) = \text{s.e.v.}\{\mathcal{E}, \mathcal{A}\mathcal{E}, \dots, \mathcal{A}^{l-1}\mathcal{E}\}$ le sous espace de Krylov par blocs associé à la paire $(\mathcal{A}, \mathcal{E})$. L'algorithme d'Arnoldi par blocs consiste à construire une base orthonormale $\{V_1, V_2, \dots, V_l\}$ de l'espace $\mathbb{K}_l(\mathcal{A}, \mathcal{E})$.

Algorithm 4 Algorithme d'Arnoldi par blocs

- Calculer la décomposition QR de $\mathcal{E} : [V_1, R] = \text{qr}(\mathcal{E})$
 - **Pour** $j = 1 : l$
 - $W = AV_j$
 - **Pour** $i = 1, \dots, j$
 - $H_{i,j}^A = V_i^T W$
 - $W = W - V_i H_{i,j}^A$
 - **Fin Pour** i
 - Calculer la décomposition QR de $W : [V_{j+1}, H_{j+1,j}^A] = \text{qr}(W)$
 - **Fin Pour** j
-

On note \mathcal{V}_l la matrice réelle de taille $m \times lr$ définie par

$$\mathcal{V}_l = [V_1, \dots, V_l]$$

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

$$\begin{aligned} \mathcal{A}\mathcal{V}_l &= \mathcal{V}_l \mathcal{H}_l^A + V_{l+1} H_{l+1,l}^A E_l^T \\ &= \mathcal{V}_{l+1} \tilde{\mathcal{H}}_l^A, \end{aligned}$$

où $E_l = [0_r, \dots, 0_r, I_r]$ est la matrice de taille $lr \times r$ composée des r dernières colonnes de la matrice identité I_{lr} et \mathcal{H}_l^A est la matrice de taille $lr \times lr$ obtenue en supprimant les r dernières lignes de $\tilde{\mathcal{H}}_l^A$.

De manière tout à fait similaire, l'application de l'algorithme d'Arnoldi par blocs à la paire $(\mathcal{B}^T, \mathcal{F})$ donne la matrice $\mathcal{W}_l \in \mathbb{R}^{n \times lr}$ et les matrices de Hessenberg, toujours notées avec les lettres \mathcal{H} et H , avec \mathcal{B} en exposant. On a ainsi les identités

$$\mathcal{B}^T \mathcal{W}_l = \mathcal{W}_l \mathcal{H}_l^{\mathcal{B}} + W_{l+1} H_{l+1,l}^{\mathcal{B}} E_l^T \quad (1.33)$$

$$= \mathcal{W}_{l+1} \tilde{\mathcal{H}}_l^{\mathcal{B}}. \quad (1.34)$$

La matrice Y_l est déterminée en imposant la condition d'orthogonalité

$$R_l := \mathcal{E}\mathcal{F}^T - \mathcal{A}X_l\mathcal{B} + X_l \perp \mathcal{L}_l(\mathcal{A}, \mathcal{B}), \quad (1.35)$$

où $\mathcal{L}_l(\mathcal{A}, \mathcal{B})$ est le sous espace vectoriel de $\mathbb{R}^{m \times n}$ constitués des matrices de la forme (1.32). Il est alors immédiat de montrer que $Y_l \in \mathbb{R}^{lr \times lr}$ est la solution de l'équation de Stein de taille réduite

$$\mathcal{H}_l^A Y_l (\mathcal{H}_l^{\mathcal{B}})^T - Y_l = \tilde{\mathcal{E}}_l \tilde{\mathcal{F}}_l^T, \quad (1.36)$$

où

$$\tilde{\mathcal{E}}_l = \mathcal{V}_l^T \mathcal{E}, \quad \text{et} \quad \tilde{\mathcal{F}}_l^T = \mathcal{F}^T \mathcal{W}_l.$$

En supposant que $\lambda_i \cdot \mu_j \neq 1$ pour toutes les valeurs propres λ_i et μ_j de \mathcal{H}_l^A et de \mathcal{H}_l^B respectivement ($i = 1, \dots, lr$, $j = 1, \dots, lr$), la solution Y_l de l'équation de Stein de taille réduite (1.36) peut être obtenue par une méthode directe comme celles qui sont décrites dans [5, 39].

Le résultat suivant donne une expression du résidu R_l qui présente l'avantage de ne pas nécessiter le calcul de l'approximation X_l .

Théorème 1.11. *Soient Y_l la solution de l'équation de taille réduite (1.36) et $X_l = \mathcal{V}_l Y_l \mathcal{W}_l^T$ la solution approchée obtenues au bout de l itérations de l'algorithme LR-ADI(2). La norme de Frobenius du résidu R_l est donnée par*

$$\|R_l\|_F = \sqrt{\alpha_l^2 + \beta_l^2 + \gamma_l^2},$$

où les réels α_l , β_l et γ_l sont définis par

$$\alpha_l = \|H_{l+1,l}^A E_l^T Y_l \mathcal{H}_l^{B,T}\|_F, \quad \beta_l = \|\mathcal{H}_l^A Y_l E_l H_{l+1,l}^{B,T}\|_F,$$

et

$$\gamma_l = \|H_{l+1,l}^A E_l^T Y_l E_l H_{l+1,l}^{B,T}\|_F.$$

Démonstration. En effet, en appliquant la condition d'orthogonalité (1.35), nous avons

$$\mathcal{V}_l^T R_l \mathcal{W}_l = O_{lr \times lr}, \quad (1.37)$$

ce qui équivaut à

$$\mathcal{V}_l^T (\mathcal{E} \mathcal{F}^T - \mathcal{A} X_l \mathcal{B} + X_l) \mathcal{W}_l = O_{lr \times lr}, \quad (1.38)$$

et donc

$$\text{et} \quad \mathcal{V}_l^T \mathcal{E} \cdot (\mathcal{W}_l^T \mathcal{F})^T - \mathcal{H}_l^A Y_l \mathcal{H}_l^{B,T} + Y_l = O_{lr \times lr}. \quad (1.39)$$

Lors de l'application de l'algorithme d'Arnoldi par blocs à la paire $(\mathcal{A}, \mathcal{E})$, le bloc initial V_1 est obtenu en calculant la décomposition QR de la matrice \mathcal{E} : $\mathcal{E} = V_1 R_1^A$. Nous pouvons donc écrire

$$\mathcal{V}_l^T \mathcal{E} = \mathcal{V}_l^T V_1 R_1^A = \tilde{E}_{lr} R_1^A, \quad (1.40)$$

où \tilde{E}_{lr} est la matrice réelle de taille $lr \times r$ dont tous les coefficients sont nuls, à l'exception du premier bloc de taille $r \times r$ qui est égal à la matrice identité $I_{r \times r}$.

De la même manière, nous avons

$$\mathcal{W}_l^T \mathcal{F} = \mathcal{W}_l^T W_1 R_1^B = \tilde{E}_{lr} R_1^B \quad (1.41)$$

En remplaçant (1.40) et (1.41) dans (1.39), nous obtenons

$$\tilde{E}_{lr} R_1^A (\tilde{E}_1 R_1^B)^T - \mathcal{H}_l^A Y_l \mathcal{H}_l^{B,T} + Y_l = 0_{lr \times lr} \quad (1.42)$$

Comme le résidu $R_l = \mathcal{E} \mathcal{F}^T - \mathcal{A} X_l \mathcal{B} + X_l$ peut être formulé comme un produit de matrices

$$R_l = [\mathcal{V}_l, V_{l+1}] \begin{pmatrix} \tilde{E}_{lr} R_1^A R_1^{B,T} \tilde{E}_{lr}^T - \mathcal{H}_l^A Y_l \mathcal{H}_l^{B,T} + Y_l & \mathcal{H}_l^A Y_l E_l H_{l+1,l}^{B,T} \\ H_{l+1,l}^A E_l^T Y_l \mathcal{H}_l^{B,T} & H_{l+1,l}^A E_l^T Y_l E_l H_{l+1,l}^{B,T} \end{pmatrix} [\mathcal{W}_l, W_{l+1}]^T, \quad (1.43)$$

la relation (1.42) nous donne

$$R_l = \mathcal{V}_{l+r} \begin{pmatrix} 0_{lr \times lr} & \mathcal{H}_l^A Y_l E_m H_{l+1,l}^{B,T} \\ H_{l+1,l}^A E_l^T Y_l \mathcal{H}_l^{B,T} & H_{l+1,l}^A E_l^T Y_l E_m H_{l+1,l}^{B,T} \end{pmatrix} \mathcal{W}_{l+r}^T. \quad (1.44)$$

En passant à la norme de Frobenius, on obtient le résultat énoncé. \square

L'intérêt pratique de ce résultat sur le résidu est de permettre de mettre en place un test d'arrêt sur l'algorithme, que nous appellerons low-rank Stein block-Arnoldi sans avoir à effectuer le produit $X_l = \mathcal{V}_l Y_l \mathcal{W}_l^T$ à chaque itération.

Le résultat suivant donne une majoration de l'erreur $X - X_l$ à la l -ième itération.

Théorème 1.12. *On suppose que $\|\mathcal{A}\|_2 < 1$ et $\|\mathcal{B}\|_2 < 1$, on note Y_l la solution de l'équation (1.36) et X_l la solution approchée de l'équation (1.31) au terme de l itérations de l'algorithme de block-Arnoldi Stein. L'erreur $X - X_l$ vérifie l'inégalité*

$$\|X - X_l\|_2 \leq \frac{\|\tilde{\mathcal{H}}_l^A Y_l E_l^T H_{l+1,l}^{B,T}\|_2 + \|H_{l+1,l}^A E_l Y_l \tilde{\mathcal{H}}_l^{B,T}\|_2 + \|H_{l+1,l}^A E_l Y_l E_l^T H_{l+1,l}^{B,T}\|_2}{1 - \|\mathcal{A}\|_2 \|\mathcal{B}\|_2}. \quad (1.45)$$

Démonstration. En multipliant l'équation (1.36) à gauche par \mathcal{V}_l et à droite par \mathcal{W}_l^T puis en appliquant les identités (1.33) et (1.33), il vient

$$[\mathcal{A} \mathcal{V}_l - V_{l+1} H_{l+1,l}^A E_l^T] Y_l [\mathcal{W}_l^T \mathcal{B} - E_l (H_{l+1,l}^B)^T W_{l+1}^T] - \mathcal{V}_l Y_l \mathcal{W}_l^T = \mathcal{V}_l \tilde{\mathcal{E}}_l \tilde{F}_l^T \mathcal{W}_l^T. \quad (1.46)$$

Or, comme on a $\mathcal{V}_l E_l = V_l$ et $\mathcal{W}_l E_l = W_l$, l'identité (1.46) devient

$$(\mathcal{A} - \mathcal{F}_l) X_l (\mathcal{B} - \mathcal{G}_l) - X_l = \mathcal{E} \mathcal{F}^T \quad (1.47)$$

où

$$\mathcal{F}_l = V_{l+1}H_{l+1,l}^A V_l^T \quad \text{et} \quad \mathcal{G}_l = W_l(H_{l+1,l}^B)^T W_{l+1}^T.$$

En soustrayant les égalités (1.31) à (1.47), nous obtenons

$$\mathcal{A}(X - X_l)\mathcal{B} - (X - X_l) = -\mathcal{A}X_l\mathcal{G}_l - \mathcal{F}_lX_l(\mathcal{B} - \mathcal{G}_l). \quad (1.48)$$

L'erreur $X_l - X$ est solution de l'équation de Stein (1.48) et peut donc s'exprimer comme sous la forme de la série (dont la convergence est assurée par les hypothèses prises sur \mathcal{A} et \mathcal{B})

$$X_l - X = \sum_{i=0}^{+\infty} \mathcal{A}^i [\mathcal{A}X_l\mathcal{G}_l + \mathcal{F}_lX_l(\mathcal{B} - \mathcal{G}_l)] \mathcal{B}^i. \quad (1.49)$$

En appliquant l'inégalité triangulaire, nous avons

$$\|\mathcal{A}X_l\mathcal{G}_l + \mathcal{F}_lX_l(\mathcal{B} - \mathcal{G}_l)\|_2 \leq \|\mathcal{A}X_l\mathcal{G}_l\|_2 + \|\mathcal{F}_lX_l\mathcal{B}\|_2 + \|\mathcal{F}_lX_l\mathcal{G}_l\|_2.$$

Or, d'après les relations issues de l'algorithme d'Arnoldi et les définitions de \mathcal{F}_l et de \mathcal{G}_l ,

$$\mathcal{A}X_l\mathcal{G}_l = \mathcal{A}V_lY_lW_l^T W_l H_{l+1,l}^{\mathcal{B},T} W_{l+1}^T \quad (1.50)$$

$$= V_{l+1}\tilde{\mathcal{H}}_l^A Y_l E_l^T H_{l+1,l}^{\mathcal{B},T} W_{l+1}^T. \quad (1.51)$$

Donc, en passant à la norme de spectrale qui est invariante par la multiplication par une matrice unitaire,

$$\|\mathcal{A}X_l\mathcal{G}_l\|_2 = \|\tilde{\mathcal{H}}_l^A Y_l E_l^T H_{l+1,l}^{\mathcal{B},T}\|_2.$$

De la même manière, on montre les identités

$$\mathcal{F}_lX_l\mathcal{B} = V_{l+1}H_{l+1,l}^A E_l Y_l \tilde{\mathcal{H}}_l^{\mathcal{B},T} W_{l+1}^T, \quad (1.52)$$

et

$$\mathcal{F}_lX_l\mathcal{G}_l = V_{l+1}H_{l+1,l}^A E_l Y_l E_l^T H_{l+1,l}^{\mathcal{B},T} W_{l+1}^T, \quad (1.53)$$

donc nous obtenons finalement la majoration

$$\begin{aligned} \|\mathcal{A}X_l\mathcal{G}_l + \mathcal{F}_lX_l(\mathcal{B} - \mathcal{G}_l)\|_2 &\leq \|\tilde{\mathcal{H}}_l^A Y_l E_l^T H_{l+1,l}^{\mathcal{B},T}\|_2 + \|H_{l+1,l}^A E_l Y_l \tilde{\mathcal{H}}_l^{\mathcal{B},T}\|_2 \\ &+ \|H_{l+1,l}^A E_l Y_l E_l^T H_{l+1,l}^{\mathcal{B},T}\|_2. \end{aligned} \quad (1.54)$$

□

Si les hypothèses du théorème (1.12) sont vérifiées, alors la borne de la norme de l'erreur donnée par la relation (1.45) est calculable et pourra être utilisée comme test d'arrêt. Nous illustrerons ce résultat de majoration de l'erreur dans la partie consacrée aux tests numériques. Remarquons aussi que dans le cas de l'équation de Lyapunov, si l est le degré du polynôme minimal de \mathcal{A} pour \mathcal{E} , $H_{l+1,l}^A$ est une matrice nulle et cela implique la convergence de l'algorithme.

Dans le cas particulier où $B = A^T$ et $E = F$, qui correspond à l'équation de Lyapunov avec $\mathcal{A} = \mathcal{A}_{\lambda,\lambda} = (A - \lambda I_m)(A + \lambda I_m)^{-1}$ et $\mathcal{B} = \mathcal{A}^T$, nous avons le résultat suivant, qui n'exige pas de conditions aussi fortes que celles du théorème précédent.

Théorème 1.13. *Supposons que $B = A^T$, en notant Y_l la solution de l'équation (1.36) et X_l la solution approchée de l'équation (1.31) obtenue au bout de la l -ème itération de l'algorithme d'Arnoldi par blocs pour l'équation de Stein. Il existe une norme matricielle $\|\cdot\|_*$ et une constante α telles que l'on ait $\|\mathcal{A}\|_* < 1$ et*

$$\|X - X_l\|_* \leq \alpha \frac{2 \|\tilde{\mathcal{H}}_l^A Y_l E_l^T H_{l+1,l}^{A,T}\|_2 + \|H_{l+1,l}^A E_l Y_l E_l^T H_{l+1,l}^{A,T}\|_2}{1 - \|\mathcal{A}\|_*^2}. \quad (1.55)$$

Démonstration. Remarquons d'abord que comme la matrice A est supposée stable, alors il est facile de vérifier que la matrice \mathcal{A} est stable au sens de Schur (c'est à dire que les valeurs propres de \mathcal{A} sont toutes dans le disque ouvert unité de \mathbb{C}). Par conséquent, il existe une norme matricielle $\|\cdot\|_*$ telle que $\|\mathcal{A}\|_* < 1$, (voir le lemme 5.6.10, Horn and Johnson [47]). Toutes les normes étant équivalentes en dimension finie, il existe une constante β telle que $\|Z\|_* \leq \alpha \|Z\|_2$ pour toute matrice Z . Donc, comme $\mathcal{B} = \mathcal{A}^T$, on a

$$H_{l+1,l}^{\mathcal{B}} = (H_{l+1,l}^{\mathcal{A}})^T.$$

Par conséquent, la relation (1.49) implique

$$\|X - X_l\|_* \leq \|\mathcal{A}X_l \mathcal{G}_l + \mathcal{F}_l X_l (\mathcal{A} - \mathcal{G}_l)\|_* \sum_{i=0}^{+\infty} \|\mathcal{A}\|_*^i.$$

Or, nous avons

$$\|\mathcal{A}X_l \mathcal{G}_l + \mathcal{F}_l X_l (\mathcal{A} - \mathcal{G}_l)\|_* \leq \alpha \|\mathcal{A}X_l \mathcal{G}_l + \mathcal{F}_l X_l (\mathcal{A} - \mathcal{G}_l)\|_2 \quad (1.56)$$

et en utilisant la majoration (1.54) avec $\mathcal{B} = \mathcal{A}^T$, nous obtenons le résultat recherché. □

Notons là aussi que si l est le degré du polynôme minimal de \mathcal{A} pour \mathcal{E} , alors le terme de droite de la majoration (1.55) est nul, ce qui implique la convergence de l'algorithme, sans condition sur la norme spectrale de \mathcal{A} .

Revenons à l'équation de Sylvester. Dans le cas où les matrices A et B sont diagonalisables, on a le résultat

Théorème 1.14. *Supposons que les matrices A et B soient diagonalisables et soient $A = PD_A P^{-1}$ et $B = QD_B Q^{-1}$ leurs décompositions spectrales. On a alors*

$$\|X - X_l\|_2 \leq C \frac{\|\tilde{\mathcal{H}}_l^A Y_l E_l^T H_{l+1,l}^{\mathcal{B},T}\|_2 + \|H_{l+1,l}^A E_l Y_l \tilde{\mathcal{H}}_l^{\mathcal{B},T}\|_2 + \|H_{l+1,l}^A E_l Y_l E_l^T H_{l+1,l}^{\mathcal{B},T}\|_2}{1 - \rho(\mathcal{A})\rho(\mathcal{B})}$$

où le réel $\kappa(P)$ désigne le conditionnement de la matrice P et $C = \kappa(P)\kappa(Q)$.

Démonstration. Comme $\mathcal{A} = \mathcal{A}_{\lambda,\mu}$ et $\mathcal{B} = \mathcal{B}_{\lambda,\mu}$, les décompositions spectrales de des matrices \mathcal{A} et \mathcal{B} sont données par

$$\mathcal{A} = P \begin{pmatrix} \frac{\alpha_1 - \lambda}{\alpha_1 + \mu} & & & \\ & \ddots & & \\ & & \frac{\alpha_m - \lambda}{\alpha_m + \mu} & \\ & & & \frac{\alpha_m - \lambda}{\alpha_m + \mu} \end{pmatrix} P^{-1} \quad \text{et} \quad \mathcal{B} = Q \begin{pmatrix} \frac{\beta_1 - \mu}{\beta_1 + \lambda} & & & \\ & \ddots & & \\ & & \frac{\beta_n - \mu}{\beta_n + \lambda} & \\ & & & \frac{\beta_n - \mu}{\beta_n + \lambda} \end{pmatrix} Q^{-1}$$

où $\{\alpha_1, \dots, \alpha_m\}$ et $\{\beta_1, \dots, \beta_n\}$ sont les valeurs propres de A et B respectivement. Donc, pour $i \geq 0$, nous avons

$$\|\mathcal{A}^i\|_2 \leq \|P\|_2 \|P^{-1}\|_2 \rho(\mathcal{A})^i = \kappa(P) \rho(\mathcal{A})^i, \quad \text{et} \quad \|\mathcal{B}^i\|_2 \leq \kappa(Q) \rho(\mathcal{B})^i. \quad (1.57)$$

En utilisant les relations (1.49) et (1.54), nous obtenons le résultat attendu. \square

Notons que, comme pour le théorème (1.12), la borne donnée par la majoration du théorème (1.14) est calculable si l'on sait estimer le rayon spectral de \mathcal{A} et celui de \mathcal{B} .

1.4.3 Forme factorisée de l'approximation de la solution

La mémoire utilisée pour le stockage est un des enjeux importants du traitement des problèmes de grande dimension. Afin d'économiser de la mémoire, il est possible d'écrire X_l sous forme factorisée. Soit $Y_l = P\Sigma Q^T$ la décomposition en valeurs singulières de la matrice Y_l , où $\Sigma \in \mathbb{R}^{lr \times lr}$ désigne la matrice des valeurs singulières de Y_l rangées dans l'ordre décroissant, P et Q étant deux matrices unitaires. Nous fixons un seuil $dtol$ et définissons P_k et Q_k les matrices constituées des k premières colonnes de P_k et de Q_k correspondant aux k valeurs singulières supérieures ou

égales à $dtol$.

En posant $\Sigma_k = \text{diag}[\sigma_1, \dots, \sigma_k]$, nous obtenons l'approximation $Y_l \approx P_k \Sigma_k Q_k^T$ (qui est la meilleure approximation de rang k de la matrice Y_l , on pourra se référer à [40] pour un exposé complet sur ce sujet).

Nous obtenons alors l'approximation

$$X_l \approx Z_l^A (Z_l^B)^T,$$

où $Z_l^A = \mathcal{V}_l Q_k \Sigma_k^{1/2}$ et $Z_l^B = \mathcal{W}_l P_k \Sigma_k^{1/2}$.

La méthode d'Arnoldi par blocs avec préconditionnement LR-ADI(2) (BA-LR-ADI(2)) est résumé dans l'algorithme suivant

Algorithm 5 Algorithme BA-LR-ADI(2)

- On fixe $X_0 = 0_{n \times s}$, l_{max} , tol_{res} , $dtol$
 - Calculer les paramètres λ et μ par l'algorithme 3.
 - Calculer $\mathcal{A}_{\lambda,\mu} = (A - \lambda I_n)(A + \mu I_n)^{-1}$, $\mathcal{B}_{\lambda,\mu} = (B - \mu I_s)(B + \lambda I_s)^{-1}$
 - Calculer $\mathcal{E}_{\lambda,\mu} = \sqrt{-(\lambda + \mu)(A + \mu I_n)^{-1}E}$ et $\mathcal{F}_{\lambda,\mu} = \sqrt{-(\lambda + \mu)(B + \lambda I_s)^{-1}F}$
 - **Pour** $l = 1 : l_{max}$
 - Mettre à jour \mathcal{V}_l , \mathcal{W}_l , \mathcal{H}_l^A et \mathcal{H}_l^B par l'algorithme 2.
 - Résoudre l'équation de Stein $\mathcal{H}_l^A Y_l (\mathcal{H}_l^B)^T - Y_l = \tilde{\mathcal{E}}_l \tilde{\mathcal{F}}_l^T$, où $\tilde{\mathcal{E}}_l = \mathcal{V}_l^T \mathcal{E}_{\lambda,\mu}$, $\tilde{\mathcal{F}}_l^T = \mathcal{F}_{\lambda,\mu}^T \mathcal{W}_l$.
 - Calculer la norme de Frobenius du résidu : $\|R_l\|_F = \sqrt{\alpha_l^2 + \beta_l^2 + \gamma_l^2}$,
 - **Si** $\|R_l\|_F < tol_{res}$ **Alors** $l = l_{max}$ **Fin Si**
 - **Fin Pour**
 - Calculer la TSVD : $Y_l \approx P_k \Sigma_k Q_k^T$; (k est le nombre de valeurs singulières supérieures à $dtol$).
 - Calculer $X_l \approx Z_l^A (Z_l^B)^T$, où $Z_l^A = \mathcal{V}_l P_k \Sigma_k^{1/2}$ et $Z_l^B = \mathcal{W}_l Q_k \Sigma_k^{1/2}$.
-

1.5 Exemples numériques

Dans ce paragraphe, nous présentons quelques exemples numériques qui illustrent des situations dans lesquelles intervient une équation de Sylvester vérifiant les conditions requises pour l'application de la méthode BA-LR-ADI(2) (block-Arnoldi low rank ADI (2) avec le choix des paramètres ADI $k_+ = 20$ et $k_- = 20$ calculés à partir de la sous-routine lp-para de LYAPACK [70]. Pour mettre en perspective l'apport de cette méthode, nous la comparons avec la méthode low rank

Sylvester block-Arnoldi (LRS-BA) [34] et Cholesky Factored ADI (fADI) [11]. Les algorithmes ont été codés en Matlab R2011a. Le critère d'arrêt pour les trois méthodes a été fixé à $\|R(X_m)\|_F \leq 10^{-7} \|E F^T\|_F$. Dans tous les tests, les coefficients des matrices E et F ont été générés par des valeurs aléatoires uniformément distribuées dans l'intervalle $[0, 1]$.

Exemple 1 Dans ce premier exemple, les matrices A et B sont obtenues par discrétisation d'une équation aux dérivées partielles stationnaire par différences finie centrées

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

sur le carré unité $[0, 1] \times [0, 1]$ avec des conditions aux bord de Dirichlet homogènes. Les tailles des matrices A et B sont $m = m_0^2$ et $n = n_0^2$ respectivement, où m_0 et n_0 sont les nombres de points de la grille intérieure dans la direction x , respectivement y . Différents choix de valeurs ont été faits pour m_0 et n_0 . Les matrices A et B étant structurées, les produits matrice-vecteur par les inverses des matrices décalées $(A + \mu I_m)$ et $(B + \lambda I_n)$ ont été effectués par l'intermédiaire de décompositions LU de ces matrices décalées.

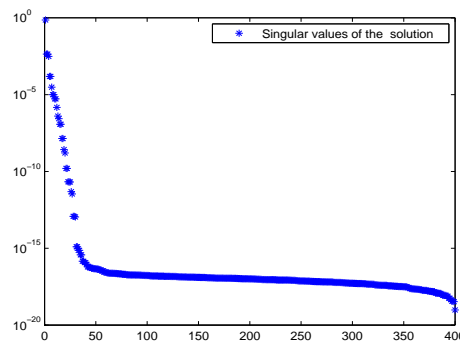


FIGURE 1.1 – Valeurs singulières de la solution exacte

La figure 1.1 représente les valeurs singulières de la solution exacte dans le cas où $m = n = 400$ 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 $\text{rang}(X) = 30$.

Dans la figure 1.2, nous donnons les courbes représentant la norme de l'erreur et sa majoration (en fonction de l'indice d'itération l) donnée par le théorème (1.12). Pour ce test aussi, nous avons choisi $m = n = 400$ et $r = 4$.

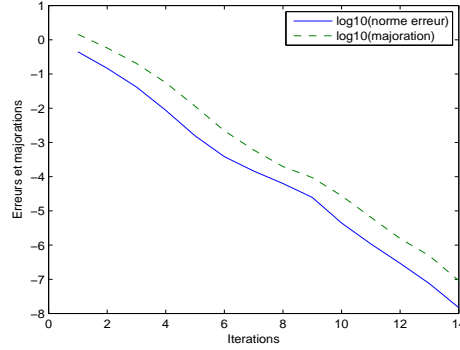


FIGURE 1.2 – norme de l'erreur et sa majoration données par (1.45)

Dans le tableau 1.1, nous donnons les résultats obtenus par les trois méthodes BA-LR-ADI(2), fADI et LRS-BA pour des matrices A et B provenant de la discrétisation des opérateurs L_A et L_B avec différentes tailles et différentes valeurs de r . Pour les trois méthodes, nous avons reporté la norme relative du résidu, le nombre total d'itérations ainsi que le temps CPU (en secondes) de calcul. Comme indiqué sur le tableau 1.1, la méthode BA-LR-ADI(2) donne les meilleurs résultats en terme de temps de calcul.

Test	Méthode	Iter.	$\ R_m\ _F / \ EF^T\ _F$	Temps
$m_0 = 30, n_0 = 30, r = 7$	LRS-BA	75	$7.08 \cdot 10^{-9}$	29.20
	fADI	20	$1.90 \cdot 10^{-10}$	23.71
	BA-LR-ADI(2)	25	$1.53 \cdot 10^{-8}$	1.90
$n_0 = 50, s_0 = 50, r = 5$	LRS-BA	105	$1.86 \cdot 10^{-8}$	43.26
	fADI	20	$7.84 \cdot 10^{-8}$	227.06
	BA-LR-ADI(2)	25	$8.30 \cdot 10^{-8}$	2.24
$n_0 = 70, s_0 = 50, r = 5$	LRS-BA	105	$1.83 \cdot 10^{-8}$	49.10
	fADI	20	$1.60 \cdot 10^{-9}$	245.06
	BA-LR-ADI(2)	40	$8.1 \cdot 10^{-8}$	7.12
$n_0 = 110, s_0 = 30, r = 4$	LRS-BA	95	$7.21 \cdot 10^{-8}$	49.10
	fADI	30	1.98	240.67
	BA-LR-ADI(2)	25	$9.11 \cdot 10^{-8}$	8.75

TABLE 1.1 – Résultats pour l'exemple 1 avec les matrices provenant de L_A et L_B .

Les résultats consignés dans le tableau 1.1 illustrent la performance de la méthode BA-LR-ADI(2) par rapport aux méthodes fADI et LRS-BA.

Exemple 2 Dans ce deuxième exemple, les matrices A et B sont issues de la collection Matrix-Market collection. Nous considérons l'équation de Sylvester

$$AX + XA - EF^T = 0,$$

rencontrée lors du calcul du grammien croisé d'un système dynamique linéaire invariant par rapport au temps. Pour plus de détail, on se référera à [7]. Dans le tableau 1.2 figurent les résultats obtenus pour différentes matrices de la collection Harwell Boeing.

Test	Méthode	Iter.	$\ R_m\ _F / \ EF^T\ _F$	Temps
$A = -\text{pde2961}$ $r = 2$	LRS-BA	180	$2.43 \cdot 10^{-8}$	20.29
	BA-LR-ADI(2)	45	$1.20 \cdot 10^{-10}$	2.25
$A = \text{sherman1}$ $r = 3$	LRS-BA	185	$6.3 \cdot 10^{-8}$	35.01
	BA-LR-ADI(2)	55	$1.80 \cdot 10^{-9}$	1.46
$A = -\text{cage9}$ $r = 5$	LRS-BA	15	$8.68 \cdot 10^{-10}$	0.42
	BA-LR-ADI(2)	15	$9.49 \cdot 10^{-15}$	12.06
$A = -\text{chem97ZtZ}$ $r = 5$	LRS-BA	50	$5.34 \cdot 10^{-8}$	3.10
	BA-LR-ADI(2)	25	$3.20 \cdot 10^{-9}$	0.92
$A = -\text{raefsky1}$ $r = 5$	LRS-BA	110	$2.47 \cdot 10^{-8}$	53.20
	BA-LR-ADI(2)	40	$9.64 \cdot 10^{-10}$	17.39

TABLE 1.2 – Résultats pour l'exemple 2 (matrices tirées de la collection Matrix-Market)

Dans le tableau 1.2, nous avons reporté les résultats obtenus par les deux méthodes LRS-BA et BA-LR-ADI(2). Comme dans le tableau 1.1, nous avons donné la norme relative du résidu, le nombre total d'itérations ainsi que le temps CPU (en secondes) de calcul. Nous n'avons pas reporté les résultats obtenus par la méthode fADI en raison de sa non convergence dans certains cas ou de sa lenteur excessive.

1.6 Conclusion

Dans ce chapitre, après avoir exposé la méthode de GMRES globale pour la résolution numérique d'équations matricielles générales, nous nous sommes intéressés au problème des équations de Sylvester (et Lyapunov) dont le second membre est factorisable sous la forme d'un produit de matrices de petit rang. Nous avons

proposé une nouvelle méthode de résolution numérique des équations de Sylvester faisant appel à une procédure de préconditionnement faisant appel à des paramètres de type ADI. Nous avons donné différents résultats de majoration de l'erreur et illustré l'efficacité de la méthode par des tests numériques.

Résolution d'une équation de type Burgers par différences finies

2.1 Introduction

Dans ce chapitre, nous nous intéressons à la résolution numérique d'une équation de type Burgers sur un domaine rectangulaire de \mathbb{R}^2 . L'équation de Burgers, qui peut être vue comme une simplification de l'équation de Navier-Stokes a été introduite en 1939 par J.M. Burgers [24] et fut originellement étudiée pour la résolution de problèmes de mécanique des fluides, en particulier dans l'étude des turbulences. Cette équation, qui est un modèle relativement simple d'équation non linéaire, intervient dans la modélisation de nombreux autres phénomènes dans des domaines aussi divers que les polymères ou l'électromagnétisme. Nous choisissons d'englober l'équation de Burgers dans une classe plus large, que nous appellerons équations de type Burgers.

Soit Ω est le domaine rectangulaire $]a, b[\times]c, d[$ inclus dans \mathbb{R}^2 , de frontière $\partial\Omega$. On désigne par $t_0 \geq 0$ et $T > t_0$ le temps initial et final respectivement. En notant $U = (u, v)$, où $u : \bar{\Omega} \rightarrow \mathbb{R}$ et $v : \bar{\Omega} \rightarrow \mathbb{R}$ désignent les fonctions inconnues, $U_0 = (u_0, v_0)$ la fonction connue donnant la condition initiale, nous considérons l'équation de type Burgers avec condition de Dirichlet

$$\left\{ \begin{array}{ll} \frac{\partial U}{\partial t} + \mu (U \cdot \nabla) U - \nu LU & = F, \quad \text{sur } \Omega \times [t_0; T] \\ U & = G, \quad \text{sur } \partial\Omega \times [t_0; T] \\ U & = U_0, \quad \text{sur } \Omega, \end{array} \right. \quad (2.1)$$

où L est un opérateur différentiel linéaire. Les fonctions $F = (f_1, f_2)$, $G = (g_1, g_2)$, u_0 et v_0 sont supposées connues. La constante strictement positive ν est appelée coefficient de viscosité et μ est un paramètre réel.

On peut écrire l'équation (2.1) de manière plus explicite sous la forme développée

$$\left\{ \begin{array}{l} \frac{\partial u}{\partial t} + \mu \left(u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} \right) - \nu Lu = f_1, \text{ sur } \Omega \times [t_0; T] \\ \frac{\partial v}{\partial t} + \mu \left(u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} \right) - \nu Lv = f_2, \text{ sur } \Omega \times [t_0; T] \\ u = g_1, \text{ sur } \partial\Omega \times [t_0; T] \\ v = g_2, \text{ sur } \partial\Omega \times [t_0; T] \\ u(., t_0) = u_0, \text{ sur } \Omega \\ v(., t_0) = v_0, \text{ sur } \Omega \end{array} \right. \quad (2.2)$$

Selon la valeur du paramètre μ on a par exemple

Equation de la chaleur	$Lu = \Delta u$ et $\mu = 0$.
Equation de Burgers	$Lu = \Delta u$ et $\mu > 0$.

Nous ne traitons pas le cas hyperbolique ($\nu = 0$), pour lequel les outils d'analyse sont plus importants que les méthodes algébriques. On pourra se référer à [4]. Il est à noter que dans le cas homogène, Hopf [46] et Cole [28] ont démontré que sous certaines conditions, l'équation de Burgers pouvait être transformée par l'intermédiaire d'un changement d'inconnue non linéaire en l'équation de la chaleur qui est linéaire et dont on connaît la solution fondamentale. Ils ont en particulier exhibé la solution analytique du système homogène

$$\left\{ \begin{array}{l} \frac{\partial U}{\partial t} + (U \cdot \nabla) U - \nu \Delta U = 0, \text{ sur } \Omega \times [t_0; T] \\ U = G, \text{ sur } \partial\Omega \times [t_0; T] \\ U(., t_0) = U_0, \text{ sur } \Omega \end{array} \right. \quad (2.3)$$

où G désigne la restriction de la solution $U = (u, v)$ à $\partial\Omega \times [t_0, T]$ qui est définie sur $\overline{\Omega} \times [t_0, T]$ par

$$\begin{aligned} u(x, y, t) &= \frac{3}{4} - \frac{1}{4 [1 + \exp((-4x + 4y - t)/32\nu)]} \\ v(x, y, t) &= \frac{3}{4} + \frac{1}{4 [1 + \exp((-4x + 4y - t)/32\nu)]}. \end{aligned} \quad (2.4)$$

Ce cas particulier est utilisé dans plusieurs articles pour valider les méthodes numériques [37, 3] développées pour la résolution de l'équation de Burgers homogène. Dans l'optique d'offrir le cadre le plus général possible, ce travail tiendra compte d'un éventuel champ de forces $f(x, y, t)$ non identiquement nul.

L'approche choisie consiste à appliquer, comme dans [3], une semi-discrétisation par différences finies. Le système d'équations différentielles ordinaires obtenu sera intégré par rapport au temps en appliquant la méthode de Runge-Kutta implicite. Cela donnera lieu à la résolution d'équations matricielles.

2.2 Schéma de résolution par différences finies

Nous considérons l'équation (2.2) sur le rectangle $\Omega = [a, b] \times [c, d]$ à laquelle nous allons appliquer la méthode des différences finies.

Soient m et n deux entiers naturels non nuls. En introduisant des subdivisions régulières $\{x_i\}_{0 \leq i \leq m+1}$ et $\{y_j\}_{0 \leq j \leq n+1}$ des intervalles $[a, b]$ et $[c, d]$ respectivement, on construit un maillage de taille $(m+2) \times (n+2)$ du rectangle $[a, b] \times [c, d]$, comme représenté figure (2.1), dont les noeuds sont les points de coordonnées (x_i, y_j) . On distingue en particulier les $m \times n$ points intérieurs $\{P_I\} = (x_i, y_j)$, où l'indice I est donné par $I = m(j-1) + 1$.

On note $h = (b-a)/(m+1)$ (respectivement $k = (d-c)/(n+1)$) les pas du maillage dans la direction de x (respectivement de y).

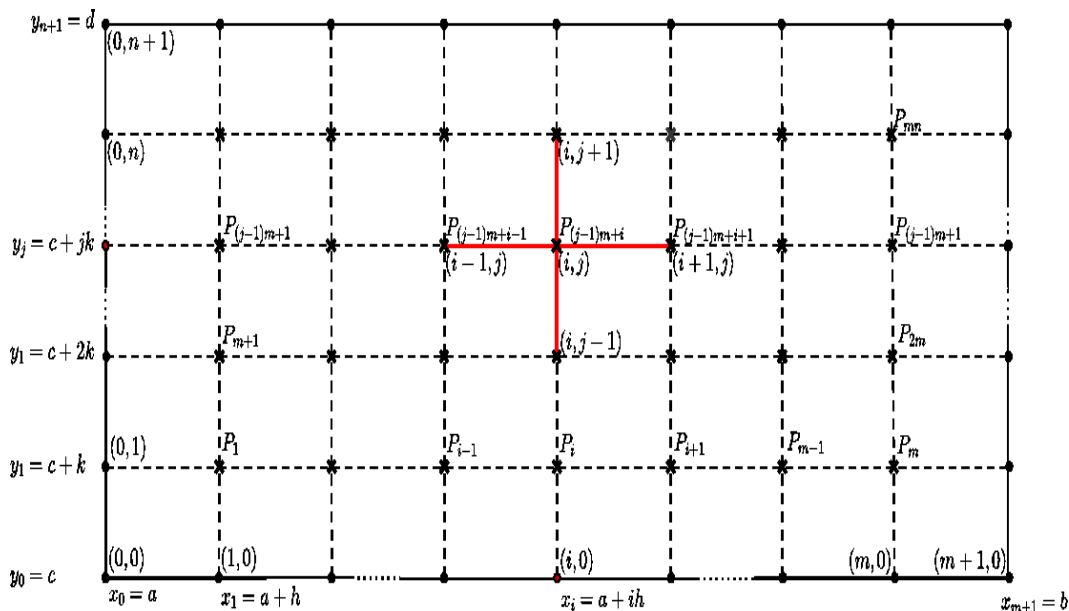


FIGURE 2.1 – maillage du rectangle

La méthode des différences finies consiste en l'approximation des dérivées spatiales grâce à des développements de Taylor par des quotients différentiels afin de calculer une valeur approchée de la solution en chacun des points intérieurs du maillage. Cette méthode mènera à l'établissement d'un système d'équations différentielles ordinaires. Supposons la fonction u suffisamment régulière, en posant $u_{i,j}(t) = u(x_i, y_j, t)$, les dérivées premières peuvent être approchées de différentes manières.

Nous utiliserons le schéma centré

$$\begin{aligned}\frac{\partial u}{\partial x}(x_i, y_j, t) &\approx \frac{u_{i+1,j}(t) - u_{i-1,j}(t)}{2h}, \\ \frac{\partial u}{\partial y}(x_i, y_j, t) &\approx \frac{u_{i,j+1}(t) - u_{i,j-1}(t)}{2k},\end{aligned}\tag{2.5}$$

et le schéma décentré (qualifié souvent de schéma "upwind")

$$\begin{aligned}\frac{\partial u}{\partial x}(x_i, y_j, t) &\approx \frac{u_{i+1,j}(t) - u_{i,j}(t)}{h}, \\ \frac{\partial u}{\partial y}(x_i, y_j, t) &\approx \frac{u_{i,j+1}(t) - u_{i,j}(t)}{k}.\end{aligned}\tag{2.6}$$

Pour discrétiser les dérivées secondes, nous utiliserons le schéma centré suivant

$$\begin{aligned}\frac{\partial^2 u}{\partial x^2}(x_i, y_j, t) &\approx \frac{u_{i+1,j}(t) - 2u_{i,j}(t) + u_{i-1,j}(t)}{h^2}, \\ \frac{\partial^2 u}{\partial y^2}(x_i, y_j, t) &\approx \frac{u_{i,j+1}(t) - 2u_{i,j}(t) + u_{i,j-1}(t)}{k^2}.\end{aligned}\tag{2.7}$$

En augmentant le degré du développement de Taylor, si le degré de régularité des fonctions le permet, on peut obtenir des approximations de dérivées d'ordre quelconque et par là-même la discrétisation de tout opérateur différentiel linéaire ou non. Pour un inventaire des différents schémas ainsi que des propriétés des schémas aux différences finies, on pourra se référer à [83]. En particulier, dans le cas de l'équation de Burgers ($L = \Delta$), l'action du Laplacien peut être approximée par

$$\Delta u_{i,j}(t) \approx \frac{u_{i+1,j}(t) - 2u_{i,j}(t) + u_{i-1,j}(t)}{h^2} + \frac{u_{i,j+1}(t) - 2u_{i,j}(t) + u_{i,j-1}(t)}{k^2}.\tag{2.8}$$

On notera $\Delta_{i,j}u(t)$ cette approximation. Soit $\mathcal{U}_{i,j}(t)$ la fonction vectorielle définie par

$$\mathcal{U}_{i,j}(t) = [u_{i,j}(t)v_{i,j}(t)]^T.$$

On introduit $\mathcal{U} : [t_0, T] \longrightarrow \mathbb{R}^{2mn}$ la fonction inconnue qui à chaque temps t associe le vecteur

$$\mathcal{U}(t) = [\mathcal{U}_{1,1}^T(t), \mathcal{U}_{2,1}^T(t), \dots, \mathcal{U}_{m,n}^T(t)]^T.$$

On applique le Laplacien discret défini par la formule (2.8) à chaque composante du vecteur $\mathcal{U}(t)$. On remarque en particulier qu'en appliquant cet opérateur en des

points intérieurs situés sur le premier rectangle le plus proche du bord (c'est à dire en des points admettant un voisin immédiat situé sur $\partial\Omega$), on observe l'apparition de termes au bord dont les valeurs sont connues car données par la connaissance de $G = U|_{\partial\Omega}$. On fait passer ces termes connus dans le second membre discrétisé en les points intérieurs de Ω . En notant $F_{i,j}(t) := [f_1(x_i, y_j, t), f_2(x_i, y_j, t)]^T \in \mathbb{R}^{2 \times 1}$ et $G_{i,j}(t) := [g_1(x_i, y_j, t), g_2(x_i, y_j, t)]^T \in \mathbb{R}^{2 \times 1}$, le second membre modifié $\phi(t)$ est explicité comme suit (il est tenu compte du coefficient $-\nu$)

$$\phi(t) = \begin{bmatrix} F_{11}(t) & +\frac{\nu}{h^2}G_{01}(t) & +\frac{\nu}{k^2}G_{10}(t) \\ F_{21}(t) & & +\frac{\nu}{k^2}G_{20}(t) \\ \vdots & & \vdots \\ F_{m-1,1}(t) & & +\frac{\nu}{k^2}G_{m-1,0}(t) \\ F_{m,1}(t) & +\frac{\nu}{h^2}G_{m+1,1}(t) & +\frac{\nu}{k^2}G_{m0}(t) \\ F_{12}(t) & +\frac{\nu}{h^2}G_{02}(t) & \\ F_{22}(t) & & \\ \vdots & & \\ F_{m-1,2}(t) & & \\ F_{m,2}(t) & +\frac{\nu}{h^2}G_{m+1,2}(t) & \\ \vdots & & \\ F_{1,n-1}(t) & +\frac{\nu}{h^2}G_{0,n-1}(t) & \\ F_{22}(t) & & \\ \vdots & & \\ F_{m-1,n-1}(t) & & \\ F_{m,n-1}(t) & +\frac{\nu}{h^2}G_{m+1,n-1}(t) & \\ F_{1,n}(t) & +\frac{\nu}{h^2}G_{0,n}(t) & +\frac{\nu}{k^2}G_{1,n}(t) \\ F_{2,n}(t) & & +\frac{\nu}{k^2}G_{2,n}(t) \\ \vdots & & \vdots \\ F_{m-1,n}(t) & & +\frac{\nu}{k^2}G_{m-1,n}(t) \\ F_{m,n}(t) & +\frac{\nu}{h^2}G_{m+1,n}(t) & +\frac{\nu}{k^2}G_{m,n}(t) \end{bmatrix}. \quad (2.9)$$

Moyennant cette manipulation consistant à relaxer les termes aux bords dans le second membre, la discrétisation du Laplacien en deux dimensions nous amène à introduire la matrice $\mathcal{D} \in \mathbb{R}^{2mn \times 2mn}$ tridiagonale par blocs donnée par

$$\mathcal{D} = \begin{bmatrix} D & J & 0 & \dots & \dots & 0 \\ J & D & J & \ddots & & \vdots \\ 0 & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & & \ddots & J & D & J \\ 0 & \dots & \dots & 0 & J & D \end{bmatrix} \quad (2.10)$$

avec

$$D = \begin{bmatrix} -\left(\frac{2}{h^2} + \frac{2}{k^2}\right) & 0 & \frac{1}{h^2} & 0 & \dots & 0 \\ 0 & -\left(\frac{2}{h^2} + \frac{2}{k^2}\right) & 0 & \frac{1}{h^2} & \vdots & \vdots \\ \frac{1}{h^2} & 0 & \ddots & \ddots & \ddots & 0 \\ 0 & \ddots & \ddots & \ddots & \ddots & \frac{1}{h^2} \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & \frac{1}{h^2} & 0 & -\left(\frac{2}{h^2} + \frac{2}{k^2}\right) \end{bmatrix} \in \mathbb{R}^{2m \times 2m} \quad (2.11)$$

et

$$J = \frac{1}{k^2} I_{2m}.$$

On a

$$\mathcal{D} \cdot \mathcal{U}(t) = [\Delta_{11}u(t), \Delta_{11}v(t), \Delta_{21}u(t), \dots, \Delta_{m1}u(t), \Delta_{m1}v(t), \Delta_{12}u(t), \dots, \Delta_{mn}v(t)]^T.$$

Passons à présent à la discrétisation de l'opérateur non linéaire $N := (U \cdot \nabla)$. Selon que l'on choisisse les différences centrées ou le schéma décentré pour l'approximation des dérivées premières, on a l'approximation

$$(\mathcal{U}_{i,j}(t) \cdot \nabla) \mathcal{U}_{i,j}(t) \approx \begin{bmatrix} u_{i,j}(t) \frac{u_{i+1,j}(t) - u_{i-1,j}(t)}{2h} + v_{i,j}(t) \frac{u_{i,j+1}(t) - u_{i,j-1}(t)}{2k} \\ u_{i,j}(t) \frac{v_{i+1,j}(t) - v_{i-1,j}(t)}{2h} + v_{i,j}(t) \frac{v_{i,j+1}(t) - v_{i,j-1}(t)}{2k} \end{bmatrix} \quad (2.12)$$

et

$$(U_{i,j}(t) \cdot \nabla)U_{i,j}(t) \approx \begin{bmatrix} u_{i,j}(t) \frac{u_{i+1,j}(t) - u_{i,j}(t)}{h} + v_{i,j}(t) \frac{u_{i,j+1}(t) - u_{i,j}(t)}{k} \\ u_{i,j}(t) \frac{v_{i+1,j}(t) - v_{i,j}(t)}{h} + v_{i,j}(t) \frac{v_{i,j+1}(t) - v_{i,j}(t)}{k} \end{bmatrix}, \quad (2.13)$$

respectivement.

On notera $N_{i,j}\mathcal{U}(t)$ cette approximation. Cette notation fait apparaître l'opérateur linéaire $\mathcal{N}(\mathcal{U}(t))$ (dont la dépendance à $\mathcal{U}(t)$ est non linéaire) appliqué au vecteur $\mathcal{U}(t)$ de la manière suivante

$$\mathcal{N}(\mathcal{U}(t)) \cdot \mathcal{U}(t) = \begin{bmatrix} \mathcal{N}_{11}(\mathcal{U}(t)) & 0_{2 \times 2} & \dots & \dots & 0_{2 \times 2} \\ 0_{2 \times 2} & \mathcal{N}_{21}(\mathcal{U}(t)) & \ddots & & \vdots \\ \vdots & \ddots & \ddots & & \vdots \\ \vdots & & \ddots & & \vdots \\ \vdots & & & \ddots & 0_{2 \times 2} \\ 0_{2 \times 2} & \dots & \dots & 0_{2 \times 2} & \mathcal{N}_{m,n}(\mathcal{U}(t)) \end{bmatrix} \cdot \begin{bmatrix} u_{11}(t) \\ v_{11}(t) \\ u_{21}(t) \\ \vdots \\ u_{mn}(t) \\ v_{mn}(t) \end{bmatrix}. \quad (2.14)$$

où les blocs diagonaux, ordonnés selon les mêmes indices que les composantes du vecteur $\mathcal{U}(t)$, sont donnés par les blocs de taille 2×2

$$N_{i,j}(\mathcal{U}(t)) = \begin{bmatrix} \frac{u_{i+1,j} - u_{i-1,j}}{2h} & \frac{u_{i,j+1} - u_{i,j-1}}{2k} \\ \frac{v_{i+1,j} - v_{i-1,j}}{2h} & \frac{v_{i,j+1} - v_{i,j-1}}{2k} \end{bmatrix}.$$

dans le cas des différences centrées, et

$$N_{i,j}(\mathcal{U}(t)) = \begin{bmatrix} \frac{u_{i+1,j} - u_{i,j}}{h} & \frac{u_{i,j+1} - u_{i,j}}{k} \\ \frac{v_{i+1,j} - v_{i,j}}{h} & \frac{v_{i,j+1} - v_{i,j}}{k} \end{bmatrix}.$$

pour le schéma décentré upwind.

Le problème est donc ramené à la résolution numérique du système d'équations différentielles ordinaires :

$$\begin{cases} \frac{d\mathcal{U}(t)}{dt} = \nu \mathcal{D} \cdot \mathcal{U}(t) - \mu \mathcal{N}(\mathcal{U}(t)) \cdot \mathcal{U}(t) + \phi(t) \\ \mathcal{U}(t_0) = \mathcal{U}_0 \end{cases} \quad (2.15)$$

où le vecteur \mathcal{U}_0 de \mathbb{R}^{2mn} est obtenu en discrétisant la fonction connue U_0 en chaque point intérieur du maillage.

2.3 Schéma de Runge-Kutta implicite

Soit donc à résoudre le système différentiel non linéaire

$$\begin{cases} \frac{d\mathcal{U}(t)}{dt} = \mathcal{F}(\mathcal{U}(t), t) \\ \mathcal{U}(t_0) = \mathcal{U}_0 \end{cases} \quad (2.16)$$

où $\mathcal{F} : \mathbb{R}^{2mn} \times [t_0, T] \rightarrow \mathbb{R}^{2mn}$ désigne la fonction qui au couple $(\mathcal{U}(t), t)$ associe le vecteur $\nu\mathcal{D} \cdot \mathcal{U}(t) - \mu\mathcal{N}(\mathcal{U}(t)) \cdot \mathcal{U}(t) + \phi(t)$.

On note \mathcal{U}_k l'approximation de la solution \mathcal{U} de l'équation (2.1) au temps $t = t_k$ au moyen de la méthode que nous exposons dans ce chapitre.

Soit s un entier naturel non nul. On introduit les coefficients de la méthode Runge-Kutta implicite à s étapes par la donnée de la matrice $\tilde{A} = (a_{ij})_{1 \leq i, j \leq s}$ et du vecteur $b = (b_1, \dots, b_s)^T$. Ces coefficients sont donnés habituellement sous la forme d'un tableau appelé tableau de Butcher [25]

$$\begin{array}{c|c} c & \tilde{A} \\ \hline & \begin{array}{ccc} a_{11} & \dots & a_{1s} \\ \vdots & \ddots & \vdots \\ a_{s1} & \dots & a_{ss} \end{array} \\ \hline & \begin{array}{ccc} b_1 & \dots & b_s \end{array} \end{array},$$

où le vecteur $c \in \mathbb{R}^s$ est choisi de telle sorte que $\tilde{A}e = c$, avec $e = (1, \dots, 1)^T \in \mathbb{R}^s$. On se réfère à [25] pour plus de détails.

Soit $t_k = t_0 + k\delta t$, $0 \leq k \leq N$ une discrétisation de l'intervalle $[t_0, T]$, où $\delta t = \frac{T - t_0}{N}$ est le pas choisi. Supposons que le vecteur \mathcal{U}_k est connu. On calcule alors \mathcal{U}_{k+1} par le schéma de Runge-Kutta implicite à s étapes en résolvant le système implicite d'inconnues les vecteurs Y_1, \dots, Y_s de taille $2mn$ suivant

$$\begin{cases} Y_1 = \mathcal{U}_k + \delta t(a_{11}\mathcal{F}(Y_1, t_k + c_1\delta t) + a_{12}\mathcal{F}(Y_2, t_k + c_2\delta t) + \dots + a_{1s}\mathcal{F}(Y_s, t_k + c_s\delta t)) \\ \vdots \\ Y_s = \mathcal{U}_k + \delta t(a_{s1}\mathcal{F}(Y_1, t_k + c_1\delta t) + a_{s2}\mathcal{F}(Y_2, t_k + c_2\delta t) + \dots + a_{ss}\mathcal{F}(Y_s, t_k + c_s\delta t)) \end{cases} \quad (2.17)$$

puis on construit le vecteur \mathcal{U}_{k+1} par la formule

$$\mathcal{U}_{k+1} = \mathcal{U}_k + \delta t(b_1\mathcal{F}(Y_1, t_k + c_1\delta t) + b_2\mathcal{F}(Y_2, t_k + c_2\delta t) + \dots + b_s\mathcal{F}(Y_s, t_k + c_s\delta t)). \quad (2.18)$$

Remarquons qu'à chaque temps t_k , les vecteurs Y_i , $i = 1, \dots, s$ dépendent de t_k mais dans un souci d'allègement des notations, nous omettrons de le mentionner. Dans les expressions (2.17) et (2.18), \mathcal{U}_k est une approximation de $\mathcal{U}(t_k)$. Le vecteur Y_i est quant à lui une approximation de la valeur intermédiaire $\mathcal{U}(t_k + c_i\delta t)$.

En notation tensorielle, les relations (2.17) et (2.18) peuvent être reformulées de la façon suivante :

$$\mathbb{Y} = e \otimes \mathcal{U}_k + \delta t(\tilde{A} \otimes I_n)\mathbb{F}(\mathbb{Y}, t_k), \quad (2.19)$$

$$\mathcal{U}_{k+1} = \mathcal{U}_k + \delta t(b^T \otimes I_{2mn})\mathbb{F}(\mathbb{Y}, t_k), \quad (2.20)$$

où I_{2mn} désigne la matrice identité de taille $2mn \times 2mn$, le vecteur \mathbb{Y} est donné par $\mathbb{Y} = \begin{bmatrix} Y_1 \\ \vdots \\ Y_s \end{bmatrix} \in \mathbb{R}^{2mns}$ et la fonction $\mathbb{F} : \mathbb{R}^{2mns} \times [t_0, T] \rightarrow \mathbb{R}^{2mns}$ est définie par :

$$\mathbb{F}(\mathbb{Y}, t) = \begin{bmatrix} \mathcal{F}(Y_1, t_k + c_1 \delta t) \\ \vdots \\ \mathcal{F}(Y_s, t_k + c_s \delta t) \end{bmatrix}.$$

On introduit la fonction vectorielle $R_k : \mathbb{R}^{2mns} \rightarrow \mathbb{R}^{2mns}$ définie par

$$R_k(\mathbb{Y}) = e \otimes \mathcal{U}_k + \delta t(\tilde{A} \otimes I_{2mn})\mathbb{F}(\mathbb{Y}, t_k) - \mathbb{Y}. \quad (2.21)$$

L'équation (2.19) est donc équivalente à l'équation non linéaire

$$R_k(\mathbb{Y}) = 0. \quad (2.22)$$

Dans la section suivante, nous proposons une méthode de résolution numérique de l'équation (2.22), dont la taille, qui est déterminée par le nombre de points choisis pour le maillage, est potentiellement importante.

2.4 Méthode quasi-Newton inexacte

Dans cette partie, nous allons donner une méthode de résolution de l'équation non linéaire (2.22) basée sur l'utilisation de la méthode de Newton. La méthode de Newton appliquée à l'équation non linéaire $R_k(\mathbb{Y}) = 0$ consiste à construire une suite $(\mathbb{Y}_p)_{p \in \mathbb{N}}$ d'approximations de la solution exacte \mathbb{Y}^* suivant le schéma suivant. On choisit un premier terme $\mathbb{Y}_0 \in \mathbb{R}^{2mns}$ et on construit les termes suivants selon la relation de récurrence

$$\mathbb{Y}_{p+1} = \mathbb{Y}_p - \mathbb{S}_p, \quad (2.23)$$

où le vecteur $\mathbb{S}_p \in \mathbb{R}^{2mns}$ est solution de l'équation linéaire d'inconnue \mathbb{X}

$$R'_{\mathbb{Y}_p}(\mathbb{X}) = R(\mathbb{Y}_p), \quad \mathbb{X} \in \mathbb{R}^{2mns} \quad (2.24)$$

dans laquelle $R'_{\mathbb{Y}_p}$ désigne la dérivée de Fréchet de R en \mathbb{Y}_p , le temps étant fixé à $t = t_k$.

La dérivée de Fréchet de R est donnée par la formule

$$R'_{\mathbb{Y}_p}(\mathbb{H}) = \delta t(\tilde{A} \otimes I_{2mn}) \text{diag}[\mathcal{F}'(t_k + c_1 \delta t, Y_{p,1}), \dots, \mathcal{F}'(t_k + c_s \delta t, Y_{p,s})] \mathbb{H} - \mathbb{H}, \quad (2.25)$$

pour tout $\mathbb{H} = [H_1, \dots, H_s]^T \in \mathbb{R}^{2mns}$, où $Y_p = [Y_{p,1}, \dots, Y_{p,s}]^T \in \mathbb{R}^{2mns}$ et où $\mathcal{F}'(t_k + c_i \delta t, Y_{p,i}) \in \mathbb{R}^{2mn \times 2mn}$ désigne la matrice jacobienne de la fonction vectorielle \mathcal{F} évaluée en $Y_{p,i}$, à l'instant $t = t_k + c_i \delta t$.

Lorsque le maillage du rectangle Ω devient très fin, les calculs de dérivée à chaque temps t_k qu'impliquent la méthode de Newton deviennent de plus en plus long, d'autant plus si le nombre d'étapes s est élevé. Il devient alors intéressant d'opter pour un schéma de type quasi-Newton où l'on remplace la différentielle $R'_{\mathbb{Y}_p}$ par une application plus simple à calculer. Pour ce faire, nous choisissons de remplacer les blocs $\mathcal{F}'(t_k + c_i \delta t, Y_{p,i})$, $1 \leq i \leq s$ par une même matrice $J_k = \mathcal{F}'(t_k, \mathcal{U}_k)$. Ainsi, l'on remplace la dérivée de Fréchet $R'_{\mathbb{Y}_p}$ par l'application linéaire $\widehat{D}_{\mathbb{Y}_p} R$ définie par

$$\begin{aligned} \widehat{D}_{\mathbb{Y}_p} R(\mathbb{H}) &= \delta t(\tilde{A} \otimes I_{2mn}) \text{diag}[J_k, \dots, J_k] \cdot \mathbb{H} - \mathbb{H} \\ &= \delta t(\tilde{A} \otimes I_{2mn})(I_s \otimes J_k) \cdot \mathbb{H} - \mathbb{H} \\ &= \delta t(\tilde{A} \otimes J_k) \cdot \mathbb{H} - \mathbb{H} \end{aligned} \quad (2.26)$$

La matrice $J_k = \mathcal{F}'(t_k, \mathcal{U}_k) \in \mathbb{R}^{2mn \times 2mn}$ peut être donnée explicitement de la façon suivante :

En prenant la dérivée de Fréchet de l'expression

$$\mathcal{F}(t_k, \mathcal{U}_k) = \nu \mathcal{D}\mathcal{U}_k + \mu \mathcal{N}(\mathcal{U}_k)\mathcal{U}_k + \phi(t_k)$$

il vient, pour tout vecteur H de \mathbb{R}^{2mn} ,

$$J_k H = \nu \mathcal{D}H - \mu D_{\mathcal{U}_k} \mathcal{N}H$$

où $D_{\mathcal{U}_k} \mathcal{N}$ est la dérivée de Fréchet de l'opérateur non linéaire \mathcal{N} qui est donnée explicitement par

$$D_{\mathcal{U}_k} \mathcal{N}(H) = \begin{bmatrix} \mathcal{A}_1(\mathcal{U}_k) & \mathcal{B}_1(\mathcal{U}_k) & 0 & \dots & 0 \\ -\mathcal{B}_2(\mathcal{U}_k) & \mathcal{A}_2(\mathcal{U}_k) & \mathcal{B}_2(\mathcal{U}_k) & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \mathcal{B}_{n-1}(\mathcal{U}_k) \\ 0 & \dots & 0 & -\mathcal{B}_n(\mathcal{U}_k) & \mathcal{A}_n(\mathcal{U}_k) \end{bmatrix}, \quad (2.27)$$

où

$$\mathcal{B}_j(\mathcal{U}_k) = \begin{bmatrix} B_{1,j}(\mathcal{U}_k) & 0 & \dots & 0 \\ 0 & B_{2,j}(\mathcal{U}_k) & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & B_{m,j}(\mathcal{U}_k) \end{bmatrix},$$

avec

$$B_{i,j}(\mathcal{U}) = \begin{bmatrix} v_{i,j}/2k & 0 \\ 0 & v_{i,j}/2k \end{bmatrix} \in \mathbb{R}^{2 \times 2},$$

et

$$\begin{bmatrix} D_{1,j} & A_{1,j} & 0 & \dots & 0 \\ -A_{2,j} & D_{2,j} & A_{2,j} & \ddots & \vdots \\ 0 & -A_{3,j} & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & A_{m-1,j} \\ 0 & \dots & 0 & -A_{m,j} & D_{m,j} \end{bmatrix}$$

avec

$$A_{i,j}(\mathcal{U}) = \begin{bmatrix} u_{i,j}/2h & 0 \\ 0 & u_{i,j}/2h \end{bmatrix} \in \mathbb{R}^{2 \times 2}.$$

Le calcul de l'incrément \mathbb{S}_p de la méthode de quasi-Newton se fera par la résolution de l'équation linéaire

$$\left[\delta t(\tilde{A} \otimes J_k) - I_{2mns} \right] \mathbb{S}_p = R(\mathbb{Y}_p). \quad (2.28)$$

Nous reformulons l'équation (2.28) en une équation matricielle. Pour cela, notons

$$\hat{\cdot} : \mathbb{R}^{2mns} \longrightarrow \mathbb{R}^{2mn \times s},$$

l'application linéaire réciproque de l'opération vec . Ainsi, pour tout vecteur \mathbb{S} de \mathbb{R}^{2mns} , on a

$$\text{vec}(\hat{\mathbb{S}}) = \mathbb{S}.$$

Proposition 2.1. *Pour tout vecteur $\mathbb{Y} = [Y_1^T, \dots, Y_s^T]^T \in \mathbb{R}^{2mns}$, où $Y_i \in \mathbb{R}^{2mn}$, en définissant pour tout t_k , $0 \leq k \leq N$, la fonction vectorielle*

$$\hat{\mathbb{F}}(\hat{\mathbb{Y}}, t_k) = [\mathcal{F}(t_k + c_1 \delta t, Y_1), \dots, \mathcal{F}(t_k + c_s \delta t, Y_s)] \in \mathbb{R}^{2mn \times s},$$

on a

$$R(\mathbb{Y}) = \text{vec}(\widehat{R}(\widehat{\mathbb{Y}}))$$

où

$$\widehat{R}(\widehat{\mathbb{Y}}) = \mathcal{U}_k e^T + \delta t \widehat{\mathbb{F}}(\widehat{\mathbb{Y}}, t_k) \widetilde{A}^T - \widehat{\mathbb{Y}}.$$

Démonstration.

Il vient immédiatement : $\text{vec}(\widehat{\mathbb{Y}}) = \mathbb{Y}$. De plus $\text{vec}(\mathcal{U}_k e^T) = \text{vec}([\mathcal{U}_k, \dots, \mathcal{U}_k]) = \mathcal{U}_k$. De la même façon, $\text{vec}(\widehat{\mathbb{F}}(\widehat{\mathbb{Y}}, t_k)) = \mathbb{F}(\mathbb{Y}, t_k)$. On vérifie enfin que

$$\text{vec}(\widehat{\mathbb{F}}(\widehat{\mathbb{Y}}, t_k) \widetilde{A}^T) = (\widetilde{A}^T \otimes I_{2mn}) \mathbb{F}(\mathbb{Y}, t_k),$$

ce qui achève la démonstration. □

De la même manière, on montre que l'équation vectorielle (2.28) est transformée en l'équation linéaire matricielle de Stein non symétrique suivante

$$(\delta t J_k) \widehat{\mathbb{S}} \widetilde{A}^T - \widehat{\mathbb{S}} = \widehat{R}(\widehat{\mathbb{Y}}). \quad (2.29)$$

En définissant l'opérateur linéaire

$$\begin{aligned} \mathcal{M} : \mathbb{R}^{2mn \times s} &\longrightarrow \mathbb{R}^{2mn \times s} \\ \widehat{\mathbb{S}} &\longmapsto (\delta t J_k) \widehat{\mathbb{S}} \widetilde{A}^T - \widehat{\mathbb{S}}, \end{aligned}$$

l'équation s'écrit

$$\mathcal{M}(\widehat{\mathbb{S}}_p) = \widehat{R}(\widehat{\mathbb{Y}}_p). \quad (2.30)$$

L'équation (2.30) est résolue numériquement par l'algorithme de GMRES global et une fois l'incrément (ou plus exactement une valeur approchée de cet incrément) $\widehat{\mathbb{S}}_p$ calculé, on obtient \mathbb{Y}_{p+1} . L'arrêt de l'exécution de l'algorithme de quasi-Newton inexact est décidé lorsque la norme $\|\widehat{\mathbb{Y}}_{p+1} - \widehat{\mathbb{Y}}_p\|_F$ est inférieure à un seuil fixé tol .

Une fois l'équation résolue, en notant $\widehat{\mathbb{Y}}$ sa solution, on calcule la valeur approchée \mathcal{U}_{k+1} de $U(t_{k+1})$ par la formule

$$\mathcal{U}_{k+1} = \mathcal{U}_k + \delta t \widehat{\mathbb{F}}(\widehat{\mathbb{Y}}, t_k) b.$$

On résume ci-dessous les étapes de la résolution de l'équation $R(\mathbb{Y})$ par la méthode de quasi-Newton.

Algorithm 6 Méthode de quasi Newton inexacte

-
- Données : t_k, \mathcal{U}_k ,
 - Choisir un premier terme $\hat{Y}_0 \in \mathbb{R}^{2mn \times s}$, fixer une tolérance tol et un nombre maximal d'itérations $kmax$
 - Initialisation : $p = 0, \hat{Y}_p = \hat{Y}_0$
 - **Tant que** [$p \leq kmax$ et erreur $> tol$]
 1. Calculer $\hat{R}(\hat{Y}_p) = \mathcal{U}_k e^T + \delta t \hat{F}(\hat{Y}_p, t_k) \tilde{A}^T - \hat{Y}_p$
 2. Résoudre l'équation de Stein $(\delta t J_k) \hat{S}_p \tilde{A}^T - \hat{S}_p = \hat{R}(\hat{Y}_p)$ par GMRES global
 3. $\hat{Y}_{p+1} = \hat{Y}_p - \hat{S}_p$
 4. $p = p + 1$, erreur = $\|\hat{Y}_{p+1} - \hat{Y}_p\|_F$
 5. $\hat{Y}_p = \hat{Y}_{p+1}$
 - **Fin du Tant que**
 - Calculer $\mathcal{U}_{k+1} = \mathcal{U}_k + \delta t \hat{F}(\hat{Y}_p, t_k) b$
-

2.5 Exemples numériques

Dans les exemples suivants, on applique la méthode décrite dans ce chapitre à des équations de type Burgers. Pour évaluer son efficacité, les exemples sont choisis de telle manière qu'une solution analytique soit connue. On donne pour chaque cas l'erreur relative

$$e_r = \frac{\sum_{1 \leq k \leq mn} \|U(P_k, T) - U_h(P_k, T)\|_2}{\sum_{1 \leq k \leq mn} \|U(P_k, T)\|_2},$$

calculée sur les $m \times n$ points intérieurs du maillage.

2.5.1 Exemple 1

On considère l'équation de la chaleur en deux dimensions sur le domaine $\Omega = [-1, 1]^2$ et l'intervalle de temps $[0, 1]$

$$\begin{cases} \frac{\partial U}{\partial t} - \Delta U = F, & \text{sur } \Omega \times [t_0; T] \\ U = G, & \text{sur } \partial\Omega \times [t_0; T] \\ U(., t_0) = U_0, & \text{sur } \Omega \end{cases} \quad (2.31)$$

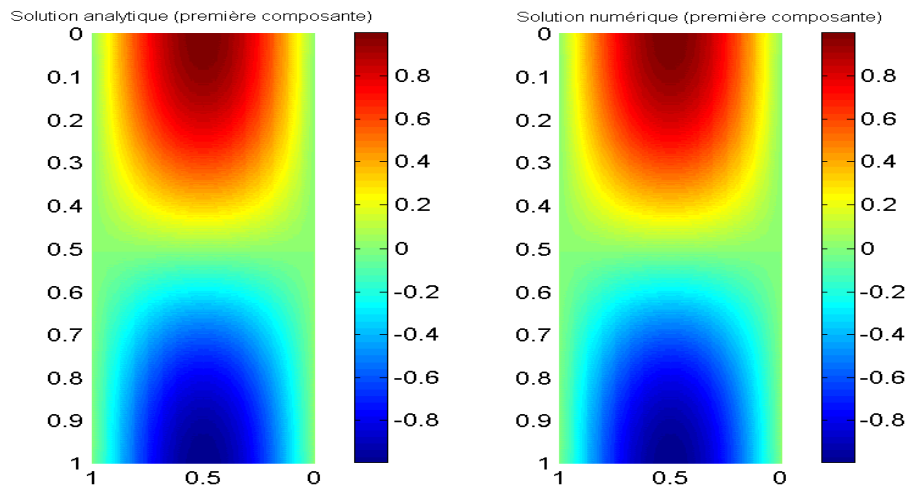
où les fonctions F et G sont choisies de façon à ce que la solution analytique soit donnée par la fonction $U = (u, v)$ définie par

$$\begin{aligned} u(x, y, t) &= \sin(\pi xt) \cos(\pi yt) \\ v(x, y, t) &= \cos(\pi xt) \sin(\pi yt). \end{aligned} \tag{2.32}$$

On représente dans les figures suivantes la première composante de la solution analytique et de la solution numérique pour un maillage de 20×20 points au temps $T = 1$, avec un pas de temps $\delta t = 5 \times 10^{-3}$ et un tableau de Butcher

$$\frac{\begin{array}{c|cc} \gamma & \gamma & 0 \\ 1-\gamma & 1-2\gamma & \gamma \end{array}}{\begin{array}{cc} 0.5 & 0.5 \end{array}}, \text{ où } \gamma = \frac{3 + \sqrt{3}}{3}.$$

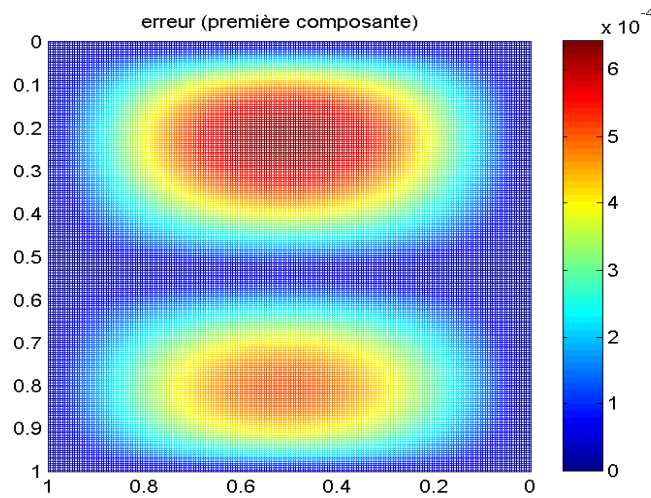
FIGURE 2.2 – première composante



Dans le cas d'un maillage à 20×20 points, l'erreur relative calculée est $e_r = 2,4 \cdot 10^{-3}$, si l'on augmente le nombre de points du maillage, on trouve une meilleure approximation ($e_r = 1,6 \cdot 10^{-3}$ pour une grille de 25×25 points), au détriment naturellement du temps de calcul.

Dans le dessin suivant, nous représentons l'erreur relative pour la première composante dans l'intérieur du domaine Ω , pour une discrétisation à 400 points et un pas de temps $\delta t = 0.005$.

FIGURE 2.3 – erreur relative



2.5.2 Exemple 2

On considère l'équation de Burgers visqueuse

$$\begin{cases} \frac{\partial U}{\partial t} + (U \cdot \nabla) U - \nu \Delta U = 0, & \text{sur } \Omega \times [t_0; T] \\ U = G, & \text{sur } \partial\Omega \times [t_0; T] \\ U(., t_0) = U_0, & \text{sur } \Omega \end{cases} \quad (2.33)$$

sur le carré unité $\bar{\Omega} = [0, 1] \times [0, 1]$ de \mathbb{R}^2 , où la fonction $G = (g_1, g_2)$ est définie de telle sorte que la solution analytique soit donnée par la formule (2.4).

Nous représentons dans les figures 2.4 à 2.7 les solutions obtenues pour $\nu = 0,01$ à différents instants : $T = 1$, puis $T = 4$. La discrétisation en espace a été faite par un schéma décentré -qui est une approche standard dans le cas de petites valeurs du paramètre de viscosité ν - en utilisant une grille de 30×30 points [3]. Pour l'intégration, nous avons utilisé le même tableau de Butcher que précédemment

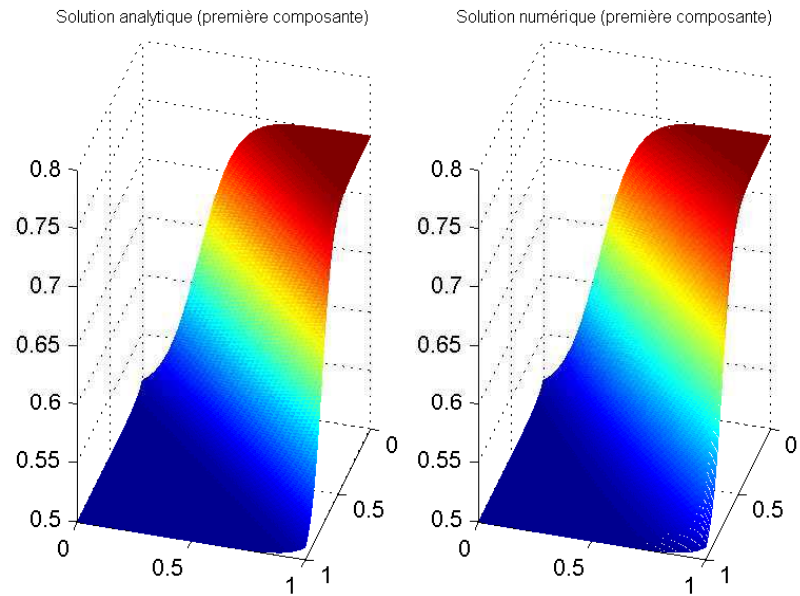


FIGURE 2.4 – Sol.analytique et numérique à l'instant $T=1$ (première composante)

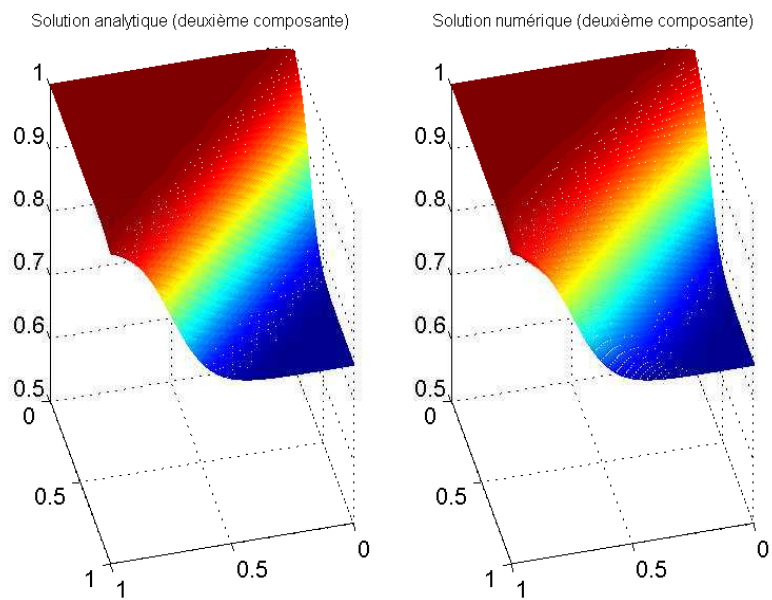
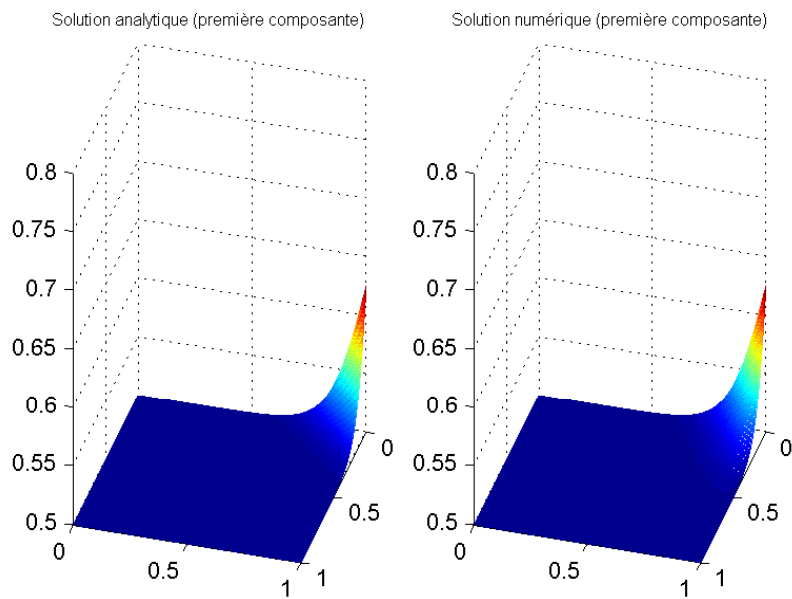
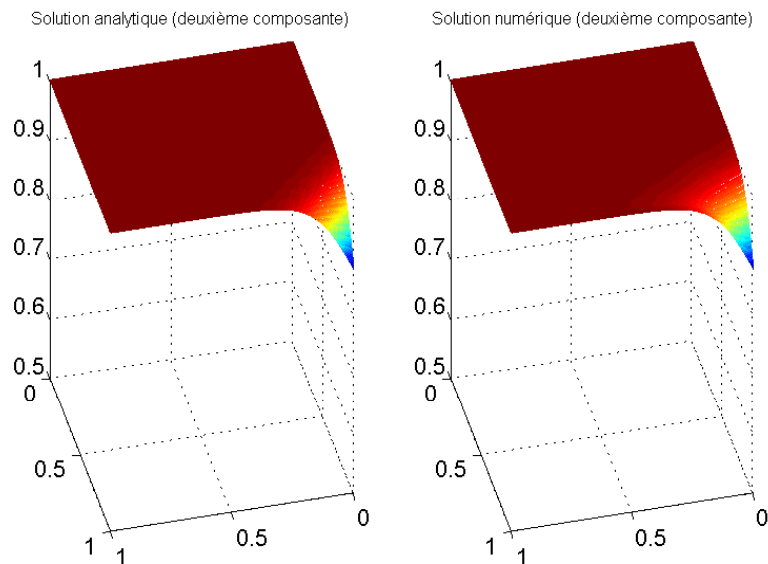


FIGURE 2.5 – Sol.analytique et numérique à l'instant $T=1$ (seconde composante)

FIGURE 2.6 – Sol.analytique et numérique à l'instant $T=4$ (première composante)FIGURE 2.7 – Sol.analytique et numérique à l'instant $T=4$ (seconde composante)

Les erreurs relatives sont respectivement de 1.4×10^{-3} à $T = 1$ et 2.1×10^{-3} à $T = 4$, pour un maillage de 30×30 points.

Dans la figure suivante, nous représentons l'erreur relative au temps $T = 1$, avec les mêmes paramètres que pour la représentation graphique de la solution.

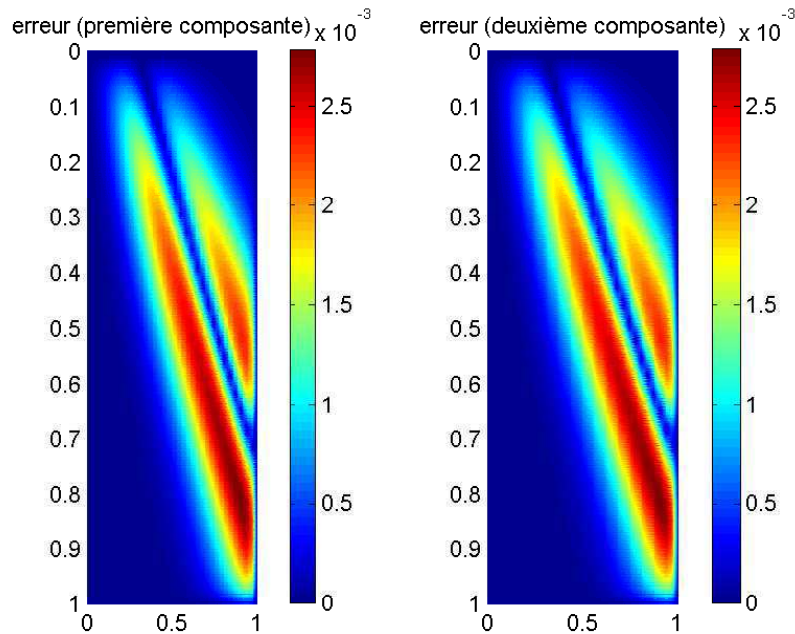


FIGURE 2.8 – Erreur relative à l'instant T=1

Dans l'article de Bahadir [3] portant sur la résolution d'équations de Burgers couplées en dimension deux, l'approche consiste en une discrétisation en espace par différences finies suivie de l'application d'un schéma d'intégration implicite, conduisant comme dans ce chapitre à la résolution d'une équation non linéaire. Les itérés de l'algorithme de Newton sont ensuite calculés numériquement par des méthodes directes de résolution de systèmes linéaires. Nous comparons dans le tableau suivant les résultats en termes de temps de calcul et de résidu au temps final sur la résolution de l'équation (2.22) obtenus par notre méthode d'une part et l'utilisation d'une méthode directe (la routine dlyap de Matlab) d'autre part, en fonction de la finesse de la discrétisation en espace. Le temps final est fixé à $T=0,2$, le pas de temps choisi est $\delta t = 0,01$ et le schéma d'intégration est le même que dans les autres exemples numériques de ce chapitre.

TABLE 2.1 – Performances comparées de GMRES global et de dlyap.

Nombre de points	GMRES global		Méthode directe	
	temps CPU	Résidu	temps CPU	Résidu
12 × 12	9.4s	1.56×10^{-5}	8.11s	1.18×10^{-5}
16 × 16	27s	1.18×10^{-5}	32.05s	1.07×10^{-5}
20 × 20	92.6s	8.94×10^{-6}	243s	9.32×10^{-6}
30 × 30	878s	8.00×10^{-6}	divergence	

Les résultats consignés dans le tableau 2.1 montrent que la méthode GMRES globale permet un traitement plus rapide et est mieux adaptée lorsque la discrétisation en espace est plus fine.

2.6 Conclusion

Dans ce chapitre, nous avons mis en oeuvre une méthode de résolution numérique d'une équation de type Burgers sur un domaine rectangulaire de \mathbb{R}^2 . La méthode de GMRES globale s'est révélée être un outil bien adapté au type d'équation linéaire matricielle à résoudre à chaque itération de l'algorithme de Newton. De plus, les résultats montrent que l'application de la méthode GMRES globale permet de résoudre les équations linéaires matricielles plus rapidement que les méthodes directes, l'avantage devenant d'autant plus visible que la discrétisation en espace est fine. Cependant, la méthode des différences finies est peu adaptée aux domaines de géométrie plus compliquée. Dans le chapitre suivant, nous allons travailler dans un cadre qui permet de s'affranchir en partie de ce problème.

Résolution d'une équation de type Burgers par une méthode sans maillage

3.1 Introduction

Les méthodes numériques les plus couramment utilisées pour la résolution des équations aux dérivées partielles sont les différences finies, les éléments finis et les volumes finis. Cependant, pour les utiliser, un important travail de préparation est nécessaire pour la génération d'un maillage du domaine sur lequel on veut appliquer ces méthodes dans le cas où la géométrie du domaine est compliquée ou que sa dimension est supérieure ou égale à 3. De plus, les expériences montrent qu'à précision comparable, les méthodes sans maillage demandent des résolutions de systèmes de taille plus petite que les méthodes d'éléments finis. Sur ce sujet particulier et plus généralement sur les bénéfices des méthodes sans maillage, on pourra se référer à [57] et aux références qui y figurent. L'utilisation de méthodes sans maillage (meshless) à l'aide de fonctions à base radiale (Radial Basis Functions) a été mise en oeuvre par EJ Kansa en 1990 [56, 55] pour la résolution d'équations aux dérivées partielles elliptiques. Dans ce chapitre, nous proposons une méthode de résolution numérique d'équations aux dérivées partielles faisant appel à une méthode sans maillage. Nous mettrons en évidence l'utilité de la méthode GMRES globale. Les cas stationnaires et évolutifs seront abordés et nous illustrerons notre méthode par des tests numériques. Rappelons tout d'abord les techniques d'interpolation par des fonctions à base radiale.

3.2 Fonctions à base radiale

Définition 3.1. Soit d un entier naturel non nul. Une fonction $\Phi : \mathbb{R}^d \rightarrow \mathbb{R}$ est dite radiale s'il existe une fonction $\phi : [0, +\infty[\rightarrow \mathbb{R}$ telle que

$$\Phi(x) = \phi(\|x\|), \text{ pour tout } x \in \mathbb{R}^d,$$

où $\|\cdot\|$ désigne une norme sur \mathbb{R}^d , que nous choisirons égale à la norme euclidienne.

Les fonctions radiales sont invariantes par les isométries vectorielles, en particulier par les rotations. La figure suivante donne deux exemples de fonctions radiales sur \mathbb{R}^2 .

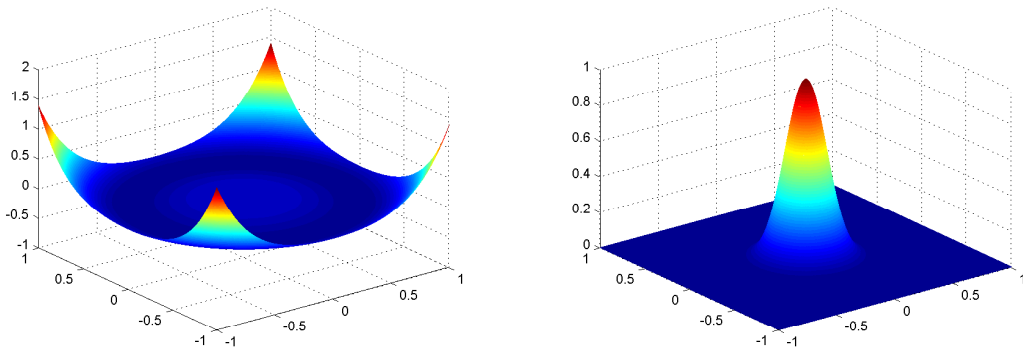


FIGURE 3.1 – à gauche : spline de type plaque mince (TPS) $x \mapsto \|x\|^2 \ln \|x\|$,
 à droite : gaussienne $x \mapsto e^{-\|x\|^2}$

Soit $\{x_i, y_i\}_{i=1, \dots, n}$ un ensemble constitué de vecteurs deux à deux distincts $x_i \in \mathbb{R}^d$ et de réels $y_i \in \mathbb{R}$. Supposons qu'il existe une fonction $f : \mathbb{R}^d \rightarrow \mathbb{R}$ telle que

$$f(x_i) = y_i, \quad 1 \leq i \leq n.$$

Interpoler f sur $\{x_i\}_{1 \leq i \leq n}$ à l'aide d'une fonction à base radiale consiste à rechercher une fonction de la forme

$$S(f)(x) = \sum_{i=1}^n \lambda_i \Phi(x - x_i),$$

(où Φ est une fonction radiale, avec $\Phi(x) = \phi(\|x\|)$ et les coefficients λ_i sont réels) qui vérifie

$$S(f)(x_i) = f(x_i) = y_i, \text{ pour chaque entier } i \text{ compris entre } 1 \text{ et } n.$$

La résolution de ce problème équivaut à celle du système linéaire

$$K\Lambda = Y, \quad (3.1)$$

où $K = [\phi(\|x_i - x_j\|)]_{1 \leq i, j \leq n}$, $\Lambda = [\lambda_1, \dots, \lambda_n]^T$ et $Y = [y_1, \dots, y_n]^T$.

Le système (3.1) n'admet pas toujours une solution. Nous verrons plus loin sous quelles conditions on est assuré de l'existence et de l'unicité d'une solution à ce problème d'interpolation. Parfois, on ajoute à l'interpolant une partie polynomiale pour s'assurer que ce problème possède bien une solution au moins dans le cas particulier de l'interpolation d'une fonction polynomiale jusqu'à un certain degré $m \geq 1$ choisi.

Notons $\Pi_{m-1}(\mathbb{R}^d)$ l'espace vectoriel des polynômes à d variables à coefficients réels de degré strictement inférieur à m . On rappelle que la dimension de $\Pi_{m-1}(\mathbb{R}^d)$ est donnée par

$$d_m = \binom{d+m-1}{d} = \frac{(d+m-1)!}{d!(m-1)!}.$$

Doté de cette précision polynomiale, l'interpolant est recherché sous la forme

$$S(f)(x) = \sum_{i=1}^n \lambda_i \Phi(x - x_i) + \sum_{j=1}^{d_m} \beta_j q_j(x), \quad (3.2)$$

où $\{q_1(x), \dots, q_{d_m}(x)\}$ est une base quelconque de $\Pi_{m-1}(\mathbb{R}^d)$.

On ajoute la condition d'orthogonalité

$$\sum_{i=1}^n \lambda_i q_j(x_i) = 0, \quad 1 \leq j \leq d_m. \quad (3.3)$$

Dans ce cas, le système d'équations linéaires à résoudre devient

$$\begin{pmatrix} K & Q \\ Q^T & O \end{pmatrix} \begin{pmatrix} \lambda \\ \beta \end{pmatrix} = \begin{pmatrix} z \\ 0 \end{pmatrix}, \quad (3.4)$$

où λ et β sont les vecteurs donnés respectivement par $\lambda = (\lambda_1, \dots, \lambda_n)^T$ et $\beta = (\beta_1, \dots, \beta_{d_m})^T$. Les matrices K et Q sont définies par

$$K = \begin{bmatrix} \phi(\|x_1 - x_1\|) & \dots & \phi(\|x_1 - x_n\|) \\ \vdots & \ddots & \vdots \\ \phi(\|x_n - x_1\|) & \dots & \phi(\|x_n - x_n\|) \end{bmatrix}, \quad Q = \begin{bmatrix} q_1(x_1) & \dots & q_{d_m}(x_1) \\ \vdots & & \vdots \\ q_1(x_n) & \dots & q_{d_m}(x_n) \end{bmatrix}.$$

et O est la matrice nulle de taille $d_m \times d_m$.

Les définitions et propositions suivantes donnent un cadre dans lequel nous serons assurés de l'existence et de l'unicité de l'interpolation par une fonction à base radiale.

Définition 3.2. Une fonction $\Phi : \mathbb{R}^d \rightarrow \mathbb{R}$ est dite conditionnellement définie positive d'ordre m sur \mathbb{R}^d si

$$\sum_{i=1}^n \sum_{j=1}^n a_i a_j \Phi(x_i - x_j) \geq 0, \quad (3.5)$$

pour tout ensemble $\{x_1, \dots, x_n\} \subset \mathbb{R}^d$ de points deux à deux distincts et tout vecteur $a = [a_1, \dots, a_n]^T \in \mathbb{R}^n$ satisfaisant, pour tout polynôme $p \in \Pi_{m-1}(\mathbb{R}^d)$, la condition suivante

$$\sum_{i=1}^n a_i p(x_i) = 0. \quad (3.6)$$

Si de plus on a l'implication (qui est en fait une équivalence)

$$\sum_{i=1}^n \sum_{j=1}^n a_i a_j \Phi(x_i - x_j) = 0 \Rightarrow a = 0,$$

alors f est dite strictement conditionnellement définie positive d'ordre m sur \mathbb{R}^d . Comme nous allons le voir, la disposition des centres x_i revêt une grande importance.

Définition 3.3. Le sous-ensemble $X = \{x_1, \dots, x_n\}$ de \mathbb{R}^d constitués d'éléments deux à deux distincts est dit Π_{m-1} -unisolvant si le seul polynôme de degré total inférieur ou égal à m s'annulant en chaque point de X est identiquement nul.

En particulier, on peut démontrer l'équivalence suivante [36]

Proposition 3.4. *L'ensemble $X = \{x_1, \dots, x_n\}$ est Π_{m-1} -unisolvant si et seulement si la matrice $P = [q_j(x_i)]_{\substack{1 \leq i \leq n \\ 1 \leq j \leq d_m}}$, où $\{q_1, \dots, q_{d_m}\}$ est une base de l'espace vectoriel $\Pi_{m-1}(\mathbb{R}^d)$ est de rang maximal.*

Nous pouvons maintenant rappeler un résultat d'existence et d'unicité d'un interpolant à base radiale

Théorème 3.5. *Soient $X = \{x_1, \dots, x_n\}$ un ensemble de points Π_{m-1} -unisolvant de \mathbb{R}^d et $y_i \in \mathbb{R}$ un ensemble de réels. Si Φ est une fonction strictement conditionnellement définie positive d'ordre m sur \mathbb{R}^d , alors, sous les conditions d'orthogonalité*

$$\forall 1 \leq k \leq d_m, \sum_{i=1}^n \lambda_i q_k(x_i) = 0, \text{ où } \{q_k\}_{1 \leq k \leq d_m} \text{ est une base de } \Pi_{m-1}(\mathbb{R}^d),$$

il existe une fonction $S(f)$ de la forme

$$S(f)(\mathbf{x}) = \sum_{i=1}^n \lambda_i \Phi(\mathbf{x} - \mathbf{x}_i) + \sum_{i=1}^{d_m} q_i(\mathbf{x})$$

qui vérifie

$$S(f)(\mathbf{x}_j) = y_j, \text{ pour } 1 \leq j \leq n.$$

Le tableau suivant donne quelques exemples de fonctions radiales strictement conditionnellement définies positives

Fonction	$\Phi_m(x)$	ordre m
multiquadriques généralisées	$(-1)^{\lceil \beta \rceil} (c^2 + \ x\ ^2)^\beta, \beta \in \mathbb{R}_+, \beta \notin \mathbb{N}$	$\lceil \beta \rceil$
puissances radiales	$(-1)^{\lceil \beta/2 \rceil} \ x\ ^\beta, \beta \in \mathbb{R}_+, \beta \notin 2\mathbb{N}$	$\lceil \beta/2 \rceil$
spline de type plaque mince (TPS)	$(-1)^{\beta+1} \ x\ ^{2\beta} \ln(\ x\), \beta \in \mathbb{N}$	$\beta + 1$
radiale sous tension	$\frac{-1}{c^3} (e^{-c\ x\ } + c\ x\)$	1

où $\lceil \lambda \rceil$ est le plus petit entier supérieur ou égal à λ et $c > 0$ est un paramètre réel. Pour plus de détails sur les fonctions radiales et leurs propriétés, on pourra se référer à [19, 20, 15, 23, 31, 32, 36, 87]. Notons aussi qu'au lieu de fonctions définies globalement sur \mathbb{R}^d , on peut envisager d'utiliser des fonctions radiales à support compact ([21, 22, 86, 89]).

Dans le paragraphe suivant, on applique l'interpolation à l'aide de fonctions à base radiale à la résolution numérique d'une équation aux dérivées partielles de type Burgers.

3.3 Méthodes sans maillage pour une EDP de type Burgers stationnaire

3.3.1 Approximation de la solution par fonctions à base radiale

Soit $\Omega \subset \mathbb{R}^d$ un ouvert borné connexe. On note $\partial\Omega$ sa frontière. On considère l'équation aux dérivées partielles

$$\begin{cases} \mu (u(\mathbf{x}) \cdot \nabla) u(\mathbf{x}) - \nu L u(\mathbf{x}) = f(\mathbf{x}), & \text{pour } \mathbf{x} \in \Omega, \\ u(\mathbf{x}) = g(\mathbf{x}), & \text{pour } \mathbf{x} \in \partial\Omega. \end{cases} \quad (3.7)$$

où L est un opérateur différentiel linéaire, $f : \Omega \rightarrow \mathbb{R}^d$ et $g : \partial\Omega \rightarrow \mathbb{R}^d$ sont deux fonctions connues $f(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_d(\mathbf{x}))$, pour tout $\mathbf{x} = (x_1, \dots, x_d) \in \Omega$ et $g(\mathbf{x}) = (g_1(\mathbf{x}), \dots, g_d(\mathbf{x}))$, pour tout $\mathbf{x} = (x_1, \dots, x_d) \in \partial\Omega$. Nous supposons que la frontière $\partial\Omega$ de Ω est suffisamment régulière pour être assuré que le problème (3.7) possède une solution $u = (u_1, \dots, u_d)$ définie sur $\overline{\Omega}$, à valeurs dans \mathbb{R}^d . La notation $(u(\mathbf{x}) \cdot \nabla) u(\mathbf{x})$ signifie

$$(u(\mathbf{x}) \cdot \nabla) u(\mathbf{x}) = \left(\sum_{i=1}^d u_i(\mathbf{x}) \frac{\partial u_1}{\partial x_i}(\mathbf{x}), \dots, \sum_{i=1}^d u_i(\mathbf{x}) \frac{\partial u_d}{\partial x_i}(\mathbf{x}) \right).$$

L'opérateur L agit sur u comme décrit ci-dessous

$$L u(\mathbf{x}) = (L u_1(\mathbf{x}), \dots, L u_d(\mathbf{x})).$$

Le réel $\nu > 0$ est appelé coefficient de viscosité. En fonction du choix du paramètre $\mu \in \mathbb{R}$ et de l'opérateur L , on a par exemple les équations suivantes

Equation de Poisson	$Lu = \Delta u$ et $\mu = 0$.
Equation biharmonique	$Lu = \Delta^2 u$ et $\mu = 0$.
Equation de Helmholtz	$Lu = \Delta u - k u$ et $\mu = 0$.
Equation de Burgers stationnaire	$Lu = \Delta u$ et $\mu > 0$.

Il est possible d'étendre cette liste en choisissant pour L n'importe quel opérateur différentiel linéaire sur \mathbb{R}^d . Dans le cas de l'équation de Helmholtz, une méthode basée sur l'utilisation de la solution fondamentale et d'interpolation par des fonctions à base radiale a été proposée en 2008 dans [17].

L'idée première est d'écrire un interpolant de la solution $u : \overline{\Omega} \rightarrow \mathbb{R}^d$ de l'équation (3.7). La fonction u étant à valeurs dans \mathbb{R}^d , on généralise les définitions du paragraphe précédent au cas vectoriel et on peut formuler le résultat

Théorème 3.6. Soient $X = \{x_1, \dots, x_n\}$ un ensemble Π_{m-1} -unisolvant de points

$$\text{de } \overline{\Omega} \subset \mathbb{R}^d \text{ et } Y = \begin{bmatrix} u_1(x_1) & \dots & u_d(x_1) \\ \vdots & & \vdots \\ u_1(x_n) & \dots & u_d(x_n) \end{bmatrix}.$$

Si Φ_m est une fonction radiale strictement conditionnellement définie positive d'ordre m sur \mathbb{R}^d , alors il existe une unique paire de matrices

$$\Lambda = \begin{bmatrix} \lambda_{1,1} & \dots & \lambda_{1,d} \\ \vdots & & \vdots \\ \lambda_{n,1} & \dots & \lambda_{n,d} \end{bmatrix} \quad \text{et} \quad \Theta = \begin{bmatrix} \theta_{1,1} & \dots & \theta_{1,d} \\ \vdots & & \vdots \\ \theta_{d_m,1} & \dots & \theta_{d_m,d} \end{bmatrix}, \quad (3.8)$$

soumises à la condition d'orthogonalité

$$\sum_{i=1}^n (\lambda_{i,1}, \dots, \lambda_{i,d}) q_j(x_i) = 0, \quad j = 1, \dots, d_m, \quad (3.9)$$

telles que la fonction $S(u) : \overline{\Omega} \rightarrow \mathbb{R}^d$ définie par

$$S(u)(x) = \sum_{i=1}^n (\lambda_{i,1}, \dots, \lambda_{i,d}) \Phi_m(x - x_i) + \sum_{j=1}^{d_m} (\theta_{j,1}, \dots, \theta_{j,d}) q_j(x), \quad \forall x \in \mathbb{R}^d, \quad (3.10)$$

où $\{q_1, \dots, q_{d_m}\}$ est une base de $\Pi_{m-1}(\mathbb{R}^d)$, vérifie

$$S(u)(x_j) = (u_1(x_j), \dots, u_d(x_j)), \quad \text{pour } 1 \leq j \leq n. \quad (3.11)$$

Les matrices Λ et Θ sont calculées en résolvant le système linéaire non singulier

$$\begin{bmatrix} K & Q \\ Q^T & O \end{bmatrix} \begin{bmatrix} \Lambda \\ \Theta \end{bmatrix} = \begin{bmatrix} Y \\ 0 \end{bmatrix}, \quad (3.12)$$

où 0 est la matrice nulle de taille $d_m \times d_m$.

Considérons les ensembles suivants

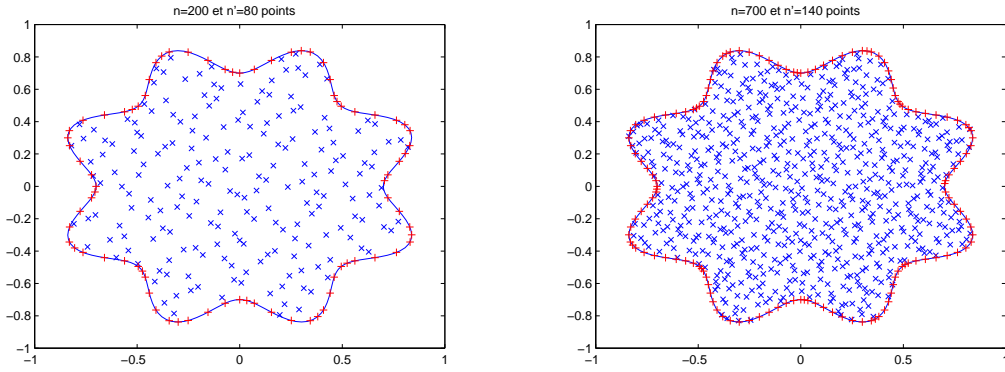
$$\begin{aligned} \mathcal{A}_n &= \{x_1, \dots, x_n\} \subset \Omega, \\ \mathcal{A}'_{n'} &= \{x'_1, \dots, x'_{n'}\} \subset \partial\Omega, \end{aligned} \quad (3.13)$$

où \mathcal{A}_n est un sous-ensemble fini de n points deux à deux distincts de Ω (points de collocation intérieurs) et $\mathcal{A}'_{n'}$ est un sous-ensemble fini de n' points deux à deux distincts de la frontière $\partial\Omega$ (points de collocations sur la frontière). Nous supposons que $\mathcal{A}'_{n'}$ contient un sous-ensemble $\Pi_{m-1}(\mathbb{R}^d)$ -unisolvant de points. Notons $N = n + n'$ le nombre total de points de collocation dans $\overline{\Omega}$ et définissons l'ensemble $\mathcal{A}_N = \mathcal{A}_n \cup \mathcal{A}'_{n'}$.

On définit le réel

$$h = \sup_{x \in \overline{\Omega}} \inf_{x' \in \mathcal{A}_N} \|x - x'\|,$$

appelé dispersion de l'ensemble \mathcal{A}_N dans $\overline{\Omega}$, qui mesure la saturation de $\overline{\Omega}$ par les points de \mathcal{A}_N : plus h est proche de 0, plus $\overline{\Omega}$ est saturé. Dans les deux figures suivantes, on visualise en comparant les deux exemples que sur la figure de droite, les points sont moins dispersés que sur celle de gauche, ce qui correspond à une valeur de h qui est plus petite.



Soit $u = (u_1, \dots, u_d) : \overline{\Omega} \rightarrow \mathbb{R}^d$ la solution analytique de l'équation (3.7) et définissons X la matrice inconnue de taille $n \times d$ donnée par

$$X = \begin{bmatrix} u_1(x_1) & \dots & u_d(x_1) \\ \vdots & & \vdots \\ u_1(x_n) & \dots & u_d(x_n) \end{bmatrix}. \quad (3.14)$$

D'après le théorème (3.6), il existe une unique fonction à base radiale u_h qui interpole la solution u sur l'ensemble \mathcal{A}_N .

Cet interpolant u_h peut s'écrire pour tout $x \in \overline{\Omega}$ comme suit

$$\begin{aligned} u_h(x) = & \sum_{i=1}^n (\lambda_{i,1}, \dots, \lambda_{i,d}) \phi_m(\|x - x_i\|) + \sum_{i=1}^{n'} (\gamma_{i,1}, \dots, \gamma_{i,d}) \phi_m(\|x' - x'_i\|) \\ & + \sum_{i=1}^{d_m} (\theta_{i,1}, \dots, \theta_{i,d}) q_i(x), \forall x \in \Omega \end{aligned} \quad (3.15)$$

soumis aux conditions d'orthogonalité

$$\sum_{i=1}^n (\lambda_{i,1}, \dots, \lambda_{i,d}) q_j(x_i) + \sum_{i=1}^{n'} (\gamma_{i,1}, \dots, \gamma_{i,d}) q_j(x'_i) = 0, \quad j = 1, \dots, d_m. \quad (3.16)$$

Nous introduisons les notations

$$\Lambda = \begin{bmatrix} \lambda_{1,1} & \cdots & \lambda_{1,d} \\ \vdots & & \vdots \\ \lambda_{n,1} & \cdots & \lambda_{n,d} \end{bmatrix}, \Gamma = \begin{bmatrix} \gamma_{1,1} & \cdots & \gamma_{1,d} \\ \vdots & & \vdots \\ \gamma_{n',1} & \cdots & \gamma_{n',d} \end{bmatrix} \text{ et } \Theta = \begin{bmatrix} \theta_{1,1} & \cdots & \theta_{1,d} \\ \vdots & & \vdots \\ \theta_{d_m,1} & \cdots & \theta_{d_m,d} \end{bmatrix}. \quad (3.17)$$

Les conditions d'interpolation couplées à celles d'orthogonalité peuvent être formulées sous forme matricielle

$$u_h(\mathbf{x}) = \lambda_m(\mathbf{x})^T \Lambda + \gamma_m(\mathbf{x})^T \Gamma + \theta_m(\mathbf{x})^T \Theta, \forall \mathbf{x} \in \Omega, \quad (3.18)$$

où

$$\lambda_m(\mathbf{x}) = \begin{bmatrix} \phi_m(\|\mathbf{x} - \mathbf{x}_1\|) \\ \vdots \\ \phi_m(\|\mathbf{x} - \mathbf{x}_n\|) \end{bmatrix}, \quad \gamma_m(\mathbf{x}) = \begin{bmatrix} \phi_m(\|\mathbf{x} - \mathbf{x}'_1\|) \\ \vdots \\ \phi_m(\|\mathbf{x} - \mathbf{x}'_{n'}\|) \end{bmatrix}, \quad \theta_m(\mathbf{x}) = \begin{bmatrix} q_1(\mathbf{x}) \\ \vdots \\ q_{d_m}(\mathbf{x}) \end{bmatrix}. \quad (3.19)$$

L'unicité de l'interpolant est équivalente à celle des matrices matrices Λ , Γ et Θ , qui sont calculées en résolvant le système linéaire non singulier suivant

$$\begin{bmatrix} K & Q \\ Q^T & O_2 \end{bmatrix} \begin{bmatrix} \Lambda \\ \Gamma \\ \Theta \end{bmatrix} = \begin{bmatrix} X \\ G \\ O_1 \end{bmatrix}, \quad (3.20)$$

où O_1 et O_2 désignent les matrices nulles de tailles $d_m \times d$ et $d_m \times d_m$ respectivement. Les matrices K et Q de tailles respectives $N \times N$ et $N \times d_m$ sont définies par

$$K = \begin{bmatrix} K_1 & M \\ M^T & K_2 \end{bmatrix}, \quad Q = \begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix}.$$

Les blocs G , K_1 , K_2 , M , Q_1 et Q_2 sont donnés par

$$\begin{aligned} K_1 &= \left[\Phi_m(\mathbf{x}_i - \mathbf{x}_j) \right]_{1 \leq i, j \leq n}, & K_2 &= \left[\Phi_m(\mathbf{x}'_i - \mathbf{x}'_j) \right]_{1 \leq i, j \leq n'}, \\ M &= \left[\Phi_m(\mathbf{x}_i - \mathbf{x}'_j) \right]_{\substack{1 \leq i \leq n \\ 1 \leq j \leq n'}}, & Q_1 &= \left[q_j(\mathbf{x}_i) \right]_{\substack{1 \leq i \leq n \\ 1 \leq j \leq d_m}}, \\ Q_2 &= \left[q_j(\mathbf{x}'_i) \right]_{\substack{1 \leq i \leq n' \\ 1 \leq j \leq d_m}}, & G &= \left[g_j(\mathbf{x}'_i) \right]_{\substack{1 \leq i \leq n' \\ 1 \leq j \leq d}}. \end{aligned} \quad (3.21)$$

On rappelle que $g = (g_1, \dots, g_d)$ est la fonction précisant la condition de Dirichlet au bord pour l'équation (2.1) et que l'ensemble $\{q_1, \dots, q_{d_m}\}$ est une base de l'espace vectoriel $\Pi_{m-1}(\mathbb{R}^d)$. Nous allons montrer comment obtenir une approximation de la fonction matricielle X .

La fonction u_h approximant la solution u sera calculée en résolvant le système linéaire non singulier (3.20).

Comme la fonction Φ_m est conditionnellement définie positive d'ordre m sur \mathbb{R}^d , les matrices $\mathbb{A} \in \mathbb{R}^{(N+d_m) \times (N+d_m)}$ et $A \in \mathbb{R}^{(n'+d_m) \times (n'+d_m)}$ respectivement définies par

$$\mathbb{A} = \begin{bmatrix} K & Q \\ Q^T & O_2 \end{bmatrix} = \begin{bmatrix} K_1 & M & Q_1 \\ M^T & K_2 & Q_2 \\ Q_1^T & Q_2^T & O_2 \end{bmatrix} \quad \text{et} \quad A = \begin{bmatrix} K_2 & Q_2 \\ Q_2^T & O_2 \end{bmatrix}, \quad (3.22)$$

sont non singulières. On se référera à [68] pour plus de détails.

Nous avons le résultat suivant

Proposition 3.7. *La matrice symétrique $S \in \mathbb{R}^{n \times n}$ donnée par*

$$S = K_1 - [M \quad Q_1] A^{-1} \begin{bmatrix} M^T \\ Q_1^T \end{bmatrix} \quad (3.23)$$

est non singulière et nous avons

$$\Lambda = S^{-1} \left(X - [M \quad Q_1] A^{-1} \tilde{G} \right) \quad \text{et} \quad \begin{bmatrix} \Gamma \\ \Theta \end{bmatrix} = A^{-1} \left(\tilde{G} - \begin{bmatrix} M^T \\ Q_1^T \end{bmatrix} \Lambda \right), \quad (3.24)$$

où Λ, Γ, Θ sont données par (3.17)-(3.20) et $\tilde{G} = \begin{bmatrix} G \\ O_1 \end{bmatrix}$.

Démonstration. Remarquons que S est le complément de Schur de la matrice A dans la matrice \mathbb{A} . Comme A et \mathbb{A} sont toutes deux non singulières, il s'ensuit que S est elle-même non singulière.

Le système linéaire (3.20) peut être reformulé comme suit

$$\begin{cases} K_1 \Lambda + M \Gamma + Q_1 \Theta = X, \\ M^T \Lambda + K_2 \Gamma + Q_2 \Theta = G, \\ Q_1^T \Lambda + Q_2^T \Gamma = O_1, \end{cases} \quad (3.25)$$

ce qui donne immédiatement

$$\begin{bmatrix} K_2 & Q_2 \\ Q_2^T & O_2 \end{bmatrix} \begin{bmatrix} \Gamma \\ \Theta \end{bmatrix} = \tilde{G} - \begin{bmatrix} M^T \\ Q_1^T \end{bmatrix} \Lambda,$$

et donc

$$\begin{bmatrix} \Gamma \\ \Theta \end{bmatrix} = A^{-1} \left(\tilde{G} - \begin{bmatrix} M^T \\ Q_1^T \end{bmatrix} \Lambda \right).$$

En substituant la dernière relation dans l'identité $K_1\Lambda + M\Gamma + Q_1\Theta = X$, nous obtenons la relation

$$\Lambda = S^{-1}\left(X - \begin{bmatrix} M & Q_1 \end{bmatrix} A^{-1}\tilde{G}\right).$$

□

La proposition 3.7 nous dit que, une fois X déterminée, la solution approchée u_h de l'équation (3.7) peut être construite. La matrice X joue donc un rôle central dans la résolution numérique de l'équation (3.7). Nous allons maintenant montrer que X est solution d'une équation matricielle non linéaire.

Introduisons quelques notations qui nous seront utiles dans la suite de notre exposé. Soient B et C les matrices données par

$$B = \begin{bmatrix} M & Q_1 \end{bmatrix} A^{-1} \quad \text{et} \quad C = A^{-1} + B^T S^{-1} B, \quad (3.26)$$

les fonctions vectorielles

$$\mathbf{a}_m(\mathbf{x}) = -S^{-1}\left(B \begin{bmatrix} \gamma_m(\mathbf{x}) \\ \theta_m(\mathbf{x}) \end{bmatrix} - \lambda_m(\mathbf{x})\right), \quad \mathbf{b}_m(\mathbf{x}) = C \begin{bmatrix} \gamma_m(\mathbf{x}) \\ \theta_m(\mathbf{x}) \end{bmatrix} - B^T S^{-1} \lambda_m(\mathbf{x}). \quad (3.27)$$

Notons $L\mathbf{a}_m$ et $L\mathbf{b}_m$ les fonctions vectorielles données par les expressions

$$\begin{aligned} L\mathbf{a}_m(\mathbf{x}) &= -S^{-1}\left(B \begin{bmatrix} L\gamma_m(\mathbf{x}) \\ L\theta_m(\mathbf{x}) \end{bmatrix} - L\lambda_m(\mathbf{x})\right), \\ L\mathbf{b}_m(\mathbf{x}) &= C \begin{bmatrix} L\gamma_m(\mathbf{x}) \\ L\theta_m(\mathbf{x}) \end{bmatrix} - B^T S^{-1} L\lambda_m(\mathbf{x}), \end{aligned} \quad (3.28)$$

où

$$L\lambda_m(\mathbf{x}) = \begin{bmatrix} L\Phi_m(\mathbf{x} - \mathbf{x}_1) \\ \vdots \\ L\Phi_m(\mathbf{x} - \mathbf{x}_n) \end{bmatrix}, \quad L\gamma_m(\mathbf{x}) = \begin{bmatrix} L\Phi_m(\mathbf{x} - \mathbf{x}'_1) \\ \vdots \\ L\Phi_m(\mathbf{x} - \mathbf{x}'_{n'}) \end{bmatrix} \quad (3.29)$$

et

$$L\theta_m(\mathbf{x}) = \begin{bmatrix} Lq_1(\mathbf{x}) \\ \vdots \\ Lq_{d_m}(\mathbf{x}) \end{bmatrix}, \quad (3.30)$$

L est l'opérateur différentiel linéaire intervenant dans l'équation (3.7).

On introduit enfin les fonctions $\nabla\mathbf{a}_m$ et $\nabla\mathbf{b}_m$ à valeurs matricielles (de tailles respectives $n \times d$ et $(n' + d_m) \times d$ respectivement) données par

$$\nabla\mathbf{a}_m(\mathbf{x}) = \left[\frac{\partial\mathbf{a}_m(\mathbf{x})}{\partial x_1}, \dots, \frac{\partial\mathbf{a}_m(\mathbf{x})}{\partial x_d} \right], \quad \nabla\mathbf{b}_m(\mathbf{x}) = \left[\frac{\partial\mathbf{b}_m(\mathbf{x})}{\partial x_1}, \dots, \frac{\partial\mathbf{b}_m(\mathbf{x})}{\partial x_d} \right]. \quad (3.31)$$

La propriété suivante sera utile pour la simplification de certaines expressions

Proposition 3.8. *On a, pour tout $i = 1, \dots, n$, $\mathbf{a}_m(\mathbf{x}_i) = \mathbf{e}_i$, où \mathbf{e}_i est le i -ème vecteur de la base canonique de \mathbb{R}^n et $\mathbf{b}_m(\mathbf{x}_i) = \mathbf{0}_n$, où $\mathbf{0}_n$ désigne le vecteur nul de \mathbb{R}^n .*

Démonstration. Nous avons d'une part

$$\begin{aligned} \begin{bmatrix} \mathbf{a}_m^T(\mathbf{x}_1) \\ \vdots \\ \mathbf{a}_m^T(\mathbf{x}_n) \end{bmatrix} &= \begin{bmatrix} \lambda_m^T(\mathbf{x}_1) - [\gamma_m^T(\mathbf{x}_1) \ \theta_m^T(\mathbf{x}_1)]B^T \\ \vdots \\ \lambda_m^T(\mathbf{x}_n) - [\gamma_m^T(\mathbf{x}_n) \ \theta_m^T(\mathbf{x}_n)]B^T \end{bmatrix} S^{-1} \\ &= \left([K_1 - [M \ Q_1]B^T] \right) S^{-1} = S.S^{-1} = I_n, \end{aligned} \quad (3.32)$$

où I_n est la matrice identité de taille $n \times n$. D'autre part, nous remarquons que

$$\begin{aligned} \begin{bmatrix} \mathbf{b}_m^T(\mathbf{x}_1) \\ \vdots \\ \mathbf{b}_m^T(\mathbf{x}_n) \end{bmatrix} &= \begin{bmatrix} [\gamma_m^T(\mathbf{x}_1) \ \theta_m^T(\mathbf{x}_1)]C - \lambda_m^T(\mathbf{x}_1)S^{-1}B \\ \vdots \\ [\gamma_m^T(\mathbf{x}_n) \ \theta_m^T(\mathbf{x}_n)]C - \lambda_m^T(\mathbf{x}_n)S^{-1}B \end{bmatrix} \\ &= [M \ Q_1]C - K_1S^{-1}B \\ &= [M \ Q_1](A^{-1} + B^TS^{-1}B) + (-S - [M \ Q_1]B^T)S^{-1}B = \mathbf{0}_{n \times n}. \end{aligned} \quad (3.33)$$

□

La proposition suivante donne une expression de u_h en fonction de X

Proposition 3.9. *La fonction u_h donnée par (3.18) peut être exprimée comme suit*

$$u_h(\mathbf{x}) = \mathbf{a}_m(\mathbf{x})^T X + \mathbf{b}_m(\mathbf{x})^T \tilde{G}, \quad (3.34)$$

et satisfait les relations

$$(u_h(\mathbf{x}) \cdot \nabla) u_h(\mathbf{x}) = \left(\mathbf{a}_m(\mathbf{x})^T X + \mathbf{b}_m(\mathbf{x})^T \tilde{G} \right) \left([\nabla \mathbf{a}_m(\mathbf{x})]^T X + [\nabla \mathbf{b}_m(\mathbf{x})]^T \tilde{G} \right), \quad (3.35)$$

et

$$Lu_h(\mathbf{x}) = [L\mathbf{a}_m(\mathbf{x})]^T X + [L\mathbf{b}_m(\mathbf{x})]^T \tilde{G}. \quad (3.36)$$

Démonstration. D'après l'expression (3.18), nous avons pour tout $\mathbf{x} \in \Omega$

$$u_h(\mathbf{x}) = \lambda_m(\mathbf{x})^T \Lambda + \gamma_m(\mathbf{x})^T \Gamma + \theta_m(\mathbf{x})^T \Theta = \lambda_m(\mathbf{x})^T \Lambda + \begin{bmatrix} \gamma_m(\mathbf{x})^T & \theta_m(\mathbf{x})^T \end{bmatrix} \begin{bmatrix} \Gamma \\ \Theta \end{bmatrix}.$$

En tenant compte de la relation (3.24), il vient

$$u_h(\mathbf{x}) = \left(\lambda_m(\mathbf{x})^T - \begin{bmatrix} \gamma_m(\mathbf{x})^T & \theta_m(\mathbf{x})^T \end{bmatrix} A^{-1} \begin{bmatrix} M^T \\ Q_1^T \end{bmatrix} \right) \Lambda + \begin{bmatrix} \gamma_m(\mathbf{x})^T & \theta_m(\mathbf{x})^T \end{bmatrix} A^{-1} \tilde{G},$$

et donc

$$u_h(\mathbf{x}) = \left(\lambda_m(\mathbf{x})^T - \begin{bmatrix} \gamma_m(\mathbf{x})^T & \theta_m(\mathbf{x})^T \end{bmatrix} B^T \right) S^{-1} (X - B\tilde{G}) + \begin{bmatrix} \gamma_m(\mathbf{x})^T & \theta_m(\mathbf{x})^T \end{bmatrix} A^{-1} \tilde{G}.$$

Ainsi, la dernière relation peut être écrite sous la forme

$$\begin{aligned} u_h(\mathbf{x}) &= \left(\lambda_m(\mathbf{x})^T - \begin{bmatrix} \gamma_m(\mathbf{x})^T & \theta_m(\mathbf{x})^T \end{bmatrix} B^T \right) S^{-1} X \\ &+ \left(\begin{bmatrix} \gamma_m(\mathbf{x})^T & \theta_m(\mathbf{x})^T \end{bmatrix} (A^{-1} + B^T S^{-1} B) - \lambda_m(\mathbf{x})^T S^{-1} B \right) \tilde{G}, \end{aligned}$$

ce qui nous donne la relation (3.71).

Soient X_j et \tilde{G}_j les vecteurs obtenus en extrayant la j -ème colonne des matrices X et \tilde{G} respectivement, pour $1 \leq j \leq d$. Alors, pour tout $1 \leq i, j \leq d$, nous avons

$$u_{h,i}(\mathbf{x}) \frac{\partial u_{h,j}(\mathbf{x})}{\partial x_i} = \left(a_m(\mathbf{x})^T X_i + b_m(\mathbf{x})^T \tilde{G}_i \right) \left(\frac{\partial a_m(\mathbf{x})^T}{\partial x_i} X_j + \frac{\partial b_m(\mathbf{x})^T}{\partial x_i} \tilde{G}_j \right),$$

où $u_h = (u_{h,1}, \dots, u_{h,d})$.

Il en découle que pour $j = 1, \dots, d$, on a

$$\begin{aligned} \sum_{i=1}^d u_{h,i}(\mathbf{x}) \frac{\partial u_{h,j}(\mathbf{x})}{\partial x_i} &= a_m(\mathbf{x})^T \left(\sum_{i=1}^d X_i \frac{\partial a_m(\mathbf{x})^T}{\partial x_i} \right) X_j + a_m(\mathbf{x})^T \left(\sum_{i=1}^d X_i \frac{\partial b_m(\mathbf{x})^T}{\partial x_i} \right) \tilde{G}_j \\ &+ b_m(\mathbf{x})^T \left(\sum_{i=1}^d \tilde{G}_i \frac{\partial a_m(\mathbf{x})^T}{\partial x_i} \right) X_j + b_m(\mathbf{x})^T \left(\sum_{i=1}^d \tilde{G}_i \frac{\partial b_m(\mathbf{x})^T}{\partial x_i} \right) \tilde{G}_j. \end{aligned}$$

Et donc pour $j = 1, \dots, d$ nous avons l'expression développée

$$\begin{aligned} \sum_{i=1}^d u_{h,i}(\mathbf{x}) \frac{\partial u_{h,j}(\mathbf{x})}{\partial x_i} &= a_m(\mathbf{x})^T X [\nabla a_m(\mathbf{x})]^T X_j + a_m(\mathbf{x})^T X [\nabla b_m(\mathbf{x})]^T \tilde{G}_j \\ &+ b_m(\mathbf{x})^T \tilde{G} [\nabla a_m(\mathbf{x})]^T X_j + b_m(\mathbf{x})^T \tilde{G} [\nabla b_m(\mathbf{x})]^T \tilde{G}_j. \end{aligned}$$

Par conséquent, $(u_h(\mathbf{x}) \cdot \nabla) u_h(\mathbf{x})$ peut être écrit sous la forme suivante

$$\begin{aligned} (u_h(\mathbf{x}) \cdot \nabla) u_h(\mathbf{x}) &= a_m(\mathbf{x})^T X [\nabla a_m(\mathbf{x})]^T X + a_m(\mathbf{x})^T X [\nabla b_m(\mathbf{x})]^T \tilde{G} \\ &+ b_m(\mathbf{x})^T \tilde{G} [\nabla a_m(\mathbf{x})]^T X + b_m(\mathbf{x})^T \tilde{G} [\nabla b_m(\mathbf{x})]^T \tilde{G}, \end{aligned}$$

et la dernière relation peut être factorisée pour obtenir l'expression (3.35). En utilisant des arguments similaires, nous obtenons la relation (3.36). \square

Considérons la fonction à valeurs matricielles définie sur $\mathbb{R}^d \times \mathbb{R}^{n \times d}$, à valeurs dans $\mathbb{R}^{1 \times d}$ donnée par la formule suivante

$$\begin{aligned} F_m(\mathbf{x}, Z) &= -f(\mathbf{x}) + \nu \left([La_m(\mathbf{x})]^T Z + [Lb_m(\mathbf{x})]^T \tilde{G} \right) \\ &\quad - \mu \left(\mathbf{a}_m(\mathbf{x})^T Z + \mathbf{b}_m(\mathbf{x})^T \tilde{G} \right) \left([\nabla \mathbf{a}_m(\mathbf{x})]^T Z + [\nabla \mathbf{b}_m(\mathbf{x})]^T \tilde{G} \right), \end{aligned}$$

et la fonction $\mathcal{F}_m : \mathbb{R}^{n \times d} \rightarrow \mathbb{R}^{n \times d}$ donnée par

$$\mathcal{F}_m(Z) = \begin{bmatrix} F_m(\mathbf{x}_1, Z) \\ \vdots \\ F_m(\mathbf{x}_n, Z) \end{bmatrix}.$$

Une application immédiate de la proposition (3.16) nous permet d'énoncer le résultat suivant

Théorème 3.10. *La fonction u_h donnée par (3.18) satisfait au schéma d'approximation*

$$\begin{cases} \mu(u_h(x) \cdot \nabla) u_h(x) - \nu Lu_h(x) = f(x), & \forall x \in \mathcal{A}_n \subset \Omega, \\ u_h(x) = g(x), & \forall x \in \mathcal{A}'_n \subset \partial\Omega. \end{cases} \quad (3.37)$$

si et seulement si la fonction matricielle X définie par (3.14) est solution de l'équation matricielle non linéaire

$$\mathcal{F}_m(X) = 0. \quad (3.38)$$

3.3.2 Calcul des coefficients matriciels

-Méthode de Newton

Nous allons résoudre numériquement l'équation matricielle non linéaire (3.38). Dans le souci de ne pas surcharger les notations à venir, nous ne ferons plus mention de l'indice m dans \mathcal{F}_m et l'équation à résoudre sera désormais notée

$$\mathcal{F}(X) = 0. \quad (3.39)$$

On va construire une suite $(X_k)_{k \geq 0}$ de matrices de taille $n \times d$ d'approximations de la solution exacte X^* de l'équation (3.39). On choisit un premier terme X_0 et, en supposant X_k connu, on définit X_{k+1} comme suit

$$X_{k+1} = X_k + S_k, \quad (3.40)$$

où la matrice $S_k \in \mathbb{R}^{n \times d}$ est solution de l'équation linéaire matricielle

$$\mathcal{F}'_{X_k}(S_k) = -\mathcal{F}(X_k). \quad (3.41)$$

L'opérateur linéaire \mathcal{F}'_{X_k} désigne la dérivée de Fréchet de \mathcal{F} en X_k dont nous allons donner une expression.

Pour $i = 1, \dots, n$, nous définissons les vecteurs

$$\begin{aligned} \mathbf{u}_i(X) &= -\nu[La_m(\mathbf{x}_i)] + \mu[\nabla a_m(\mathbf{x}_i)] \left(X^T a_m(\mathbf{x}_i) + \begin{bmatrix} G^T & O_1^T \end{bmatrix} \mathbf{b}_m(\mathbf{x}_i) \right) \\ &= -\nu[La_m(\mathbf{x}_i)] + \mu[\nabla a_m(\mathbf{x}_i)] X^T e_i, \end{aligned} \quad (3.42)$$

et

$$\mathbf{v}_i = a_m(\mathbf{x}_i),$$

où e_i est le i -ème vecteur de la base canonique de \mathbb{R}^n .

Soit $U(X)$ la matrice de taille $n \times n$ définie par $U(X) = [\mathbf{u}_1(X), \dots, \mathbf{u}_n(X)]$ et $V = [\mathbf{v}_1, \dots, \mathbf{v}_n]$. Pour $i = 1, \dots, n$, posons

$$E_i = \text{diag}(e_i) \quad \text{et} \quad A_i = E_i V^T. \quad (3.43)$$

On définit les matrices $B_i(X)$, pour $i = 1, \dots, n$ de taille $d \times d$ données par

$$B_i(X) = \mu \left([\nabla a_m(\mathbf{x}_i)]^T X + [\nabla b_m(\mathbf{x}_i)]^T \begin{bmatrix} G \\ O_1 \end{bmatrix} \right), \quad (3.44)$$

et enfin on définit la matrice de taille $nd \times nd$ définie par blocs par

$$B(X) = \begin{bmatrix} B_1(X) & 0 & \dots & 0 \\ 0 & \ddots & & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & B_n(X) \end{bmatrix}.$$

Ainsi, la dérivée de Fréchet recherchée \mathcal{F}'_X est donnée de façon immédiate par la formule

$$\mathcal{F}'_X(H) = \begin{bmatrix} \mathbf{u}_1(X)^T \\ \vdots \\ \mathbf{u}_n(X)^T \end{bmatrix} H + \begin{bmatrix} \mathbf{v}_1^T H B_1(X) \\ \vdots \\ \mathbf{v}_n^T H B_n(X) \end{bmatrix}. \quad (3.45)$$

À chaque étape de l'algorithme de Newton, il est nécessaire de mettre à jour la dérivée de Fréchet qui dépend de X_k . Afin de limiter le temps de calcul qui peut être très important, d'autant plus lorsque le nombre de points de collocation est important, nous allons commencer par donner une nouvelle écriture de \mathcal{F}'_{X_k} qui tire profit des facteurs invariants par rapport à X_k .

Proposition 3.11. *La dérivée de Fréchet de \mathcal{F} satisfait les relations*

$$\text{vec}(\mathcal{F}'_X(H)^T) = \text{vec}(H^T U(X)) + B(X)^T \text{vec}(H^T V), \quad (3.46)$$

et

$$\mathcal{F}'_X(H) = U(X)^T H + \sum_{i=1}^n A_i H B_i(X), \quad (3.47)$$

où A_i et B_i , $i = 1, \dots, n$ sont données par (3.43) et (3.44) respectivement.

Démonstration. Nous avons les relations

$$\text{vec}\left(\begin{bmatrix} u_1(X)^T H \\ \vdots \\ u_n(X)^T H \end{bmatrix}\right)^T = \begin{bmatrix} H^T u_1(X) \\ \vdots \\ H^T u_n(X) \end{bmatrix} = (I_n \otimes H^T) \text{vec}(U(X)), \quad (3.48)$$

et

$$\text{vec}\left(\begin{bmatrix} v_1^T H B_1(X) \\ \vdots \\ v_n^T H B_n(X) \end{bmatrix}\right)^T = \begin{bmatrix} B_1^T(X) H^T v_1 \\ \vdots \\ B_n^T(X) H^T v_n \end{bmatrix} = B(X)^T (I_n \otimes H^T) \text{vec}(V). \quad (3.49)$$

La relation (3.46) est obtenue en appliquant l'identité

$$\text{vec}(AXB) = (B^T \otimes A) \text{vec}(X).$$

On remarque que $B(X)$ peut s'écrire

$$B(X) = \sum_{i=1}^n E_i \otimes B_i(X).$$

Donc, en utilisant les propriétés du produit de Kronecker, en particulier

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

nous obtenons

$$\text{vec}(\mathcal{F}'_X(H)^T) = (I_n \otimes H^T) \text{vec}(U(X)) + \left(\sum_{i=1}^n E_i \otimes (H B_i(X))^T\right) \text{vec}(V).$$

Il vient alors

$$\text{vec}(\mathcal{F}'_X(H)^T) = \text{vec}(H^T U(X) + \sum_{i=1}^n (H B_i(X))^T V E_i),$$

et

$$\mathcal{F}'_X(H) = U(X)^T H + \sum_{i=1}^n E_i V^T H B_i(X),$$

ce qui nous permet de conclure. \square

La relation (3.47) montre qu'à chaque étape de la méthode de Newton, il faut résoudre une équation de Sylvester généralisée à $n + 1$ termes, ce qui du point de vue numérique est difficilement envisageable, d'autant que n est un nombre relativement important. Nous retenons la formulation (3.46) pour la suite des calculs.

Proposition 3.12. *L'application*

$$\begin{aligned} \mathcal{F}' : \mathbb{R}^{n \times d} &\longrightarrow \mathcal{L}(\mathbb{R}^{n \times d}, \mathbb{R}^{n \times d}) \\ X &\mapsto \mathcal{F}'(X) = \mathcal{F}'_X \end{aligned}$$

où \mathcal{F}'_X est la dérivée de Fréchet de \mathcal{F} en X est lipschitzienne.

Démonstration. Considérons les matrices X_1 , X_2 et H de taille $n \times d$ et à coefficients réels. D'après l'égalité (3.45) nous avons

$$\begin{aligned} (\mathcal{F}'(X_1) - \mathcal{F}'(X_2))(H) &= \mu \begin{bmatrix} \mathbf{a}_m^T(\mathbf{x}_1)(X_1 - X_2)\nabla\mathbf{a}_m^T(\mathbf{x}_1) \\ \vdots \\ \mathbf{a}_m^T(\mathbf{x}_n)(X_1 - X_2)\nabla\mathbf{a}_m^T(\mathbf{x}_n) \end{bmatrix} H \\ &+ \mu \begin{bmatrix} \mathbf{a}_m^T(\mathbf{x}_1)H\nabla\mathbf{a}_m^T(\mathbf{x}_1) \\ \vdots \\ \mathbf{a}_m^T(\mathbf{x}_n)H\nabla\mathbf{a}_m^T(\mathbf{x}_n) \end{bmatrix} (X_1 - X_2). \end{aligned}$$

Remarquons que

$$\begin{bmatrix} \mathbf{a}_m^T(\mathbf{x}_1)(X_1 - X_2)\nabla\mathbf{a}_m^T(\mathbf{x}_1) \\ \vdots \\ \mathbf{a}_m^T(\mathbf{x}_n)(X_1 - X_2)\nabla\mathbf{a}_m^T(\mathbf{x}_n) \end{bmatrix} = \mathcal{V}^T (I_n \otimes (X_1 - X_2)) \mathcal{A} \quad (3.50)$$

et

$$\begin{bmatrix} \mathbf{a}_m^T(\mathbf{x}_1)H\nabla\mathbf{a}_m^T(\mathbf{x}_1) \\ \vdots \\ \mathbf{a}_m^T(\mathbf{x}_n)H\nabla\mathbf{a}_m^T(\mathbf{x}_n) \end{bmatrix} = \mathcal{V}^T (I_n \otimes H) \mathcal{A}, \quad (3.51)$$

où

$$\mathcal{V} = \text{diag}(v_1, \dots, v_n) = \begin{bmatrix} \mathbf{a}_m(\mathbf{x}_1) & 0 & \dots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & \mathbf{a}_m(\mathbf{x}_n) \end{bmatrix} \quad \text{et} \quad \mathcal{A} = \begin{bmatrix} \nabla\mathbf{a}_m^T(\mathbf{x}_1) \\ \vdots \\ \nabla\mathbf{a}_m^T(\mathbf{x}_n) \end{bmatrix}.$$

Par conséquent, nous avons

$$(\mathcal{F}'(X_1) - \mathcal{F}'(X_2))(H) = \mu \mathcal{V}^T (I_n \otimes (X_1 - X_2)) \mathcal{A}H + \mu \mathcal{V}^T (I_n \otimes H) \mathcal{A}(X_1 - X_2).$$

En passant à la norme de Frobenius, qui est une norme matricielle, nous obtenons

$$\begin{aligned} & \|(\mathcal{F}'(X_1) - \mathcal{F}'(X_2))(H)\|_F \leq \\ & \mu \|\mathcal{A}\|_F \|\mathcal{V}\|_F \left(\|I_n \otimes (X_1 - X_2)\|_F \|H\|_F + \|I_n \otimes H\|_F \|X_1 - X_2\|_F \right). \end{aligned}$$

Donc, comme $\|I_n \otimes (X_1 - X_2)\|_F = \sqrt{n} \|X_1 - X_2\|_F$ et $\|I_n \otimes H\|_F = \sqrt{n} \|H\|_F$, il vient

$$\|(\mathcal{F}'(X_1) - \mathcal{F}'(X_2))(H)\|_F \leq 2\mu\sqrt{n} \|\mathcal{A}\|_F \|\mathcal{V}\|_F \|X_1 - X_2\|_F \|H\|_F.$$

En prenant le maximum de $\frac{\|(\mathcal{F}'(X_1) - \mathcal{F}'(X_2))(H)\|_F}{\|H\|_F}$ pour $\|H\|_F = 1$, nous pouvons conclure que

$$\|\mathcal{F}'(X_1) - \mathcal{F}'(X_2)\| \leq 2\mu\sqrt{n} \|\mathcal{A}\|_F \|\mathcal{V}\|_F \|X_1 - X_2\|_F,$$

ce qui achève la démonstration. \square

Le calcul de l'incrément S_k à chaque étape de l'algorithme de Newton nécessite la résolution de l'équation matricielle linéaire (3.41). La taille du problème dépend du nombre total N de points de collocation choisis. De plus, comme l'action de la dérivée de Fréchet de \mathcal{F} ne peut pas être identifiée à une multiplication par une matrice, la méthode Global-GMRES exposée dans le premier chapitre se prête particulièrement bien à la résolution de ce problème. Nous l'appliquons dans sa version avec redémarrage.

-Méthode de Newton inexacte GMRES global

En appliquant l'algorithme GMRES global, l'équation matricielle (3.38) n'est généralement pas résolue de manière exacte et c'est en fait une méthode de Newton inexacte qui est appliquée à l'équation (3.39). Cette méthode est décrite ci-dessous comme suit

On choisit un premier terme X_0 et on définit à l'étape $k + 1$ l'approximation X_{k+1}

$$X_{k+1} = X_k + S_k \tag{3.52}$$

où S_k est la solution exacte de l'équation matricielle linéaire perturbée

$$\mathcal{F}'_{X_k}(S_k) = -\mathcal{F}(X_k) + R_k. \tag{3.53}$$

avec R_k le résidu de l'équation linéaire (3.38) évalué en S_k .

À chaque itération de l'algorithme de Newton inexact (3.53), on résout numériquement l'équation matricielle par l'algorithme Global-GMRES. Etant donnée la taille potentielle du problème, on ne cherche évidemment pas à résoudre exactement cette équation. On impose des conditions relaxées de la forme

$$\|R_k\|_F \leq \delta_k \|\mathcal{F}(X_k)\|_F, \quad (3.54)$$

où le facteur δ_k peut être choisi de différentes manières, ce qui permet d'améliorer la convergence; on pourra se référer à [33]. Cette possibilité est importante parce que la condition (3.54) est utilisée dans le test d'arrêt de l'algorithme Global-GMRES appliqué à l'équation (1.1) qui porte sur le résidu de taille réduite. Comme $R_k = \mathcal{F}'_{X_k}(S_k) + \mathcal{F}(X_k)$ est le résidu exact de (1.1), chaque δ_k reflète la précision avec laquelle S_k résout (1.1). D'après le théorème 6.1.2 dans [58, p.96], nous avons le résultat portant sur la convergence

Théorème 3.13. *Soit X^* une solution de (3.39), alors il existe deux réels $\eta > 0$ et $\delta > 0$ tels que si X_0 est choisi de telle manière que $\|X^* - X_0\|_F \leq \eta$, alors, si pour tout $k \geq 0$, il existe un facteur positif δ_k ($\delta_k \in]0, \delta]$) tel que*

$$\|R_k\|_F \leq \delta_k \|\mathcal{F}(X_k)\|_F, \quad (3.55)$$

on a

1. Les itérés (X_k) produits par l'algorithme de Newton inexact converge linéairement vers X^* .
2. La convergence est superlinéaire si $\lim_{k \rightarrow \infty} (\delta_k) = 0$.
3. S'il existe $K_\delta > 0$ tel que $\delta_k \leq K_\delta \|\mathcal{F}(X_k)\|_F$, alors la convergence est quadratique :

$$\|X_{k+1} - X^*\|_F = O(\|X_k - X^*\|_F^2).$$

Remarque 3.14. L'opérateur linéaire qui apparaît dans chaque itération de l'algorithme de Newton peut avoir de nombreuses valeurs singulières proches de l'origine. Dans ce cas, le problème est mal conditionné et demande l'utilisation de techniques de régularisation comme par exemple la méthode de Tikhonov [16].

Les étapes de la méthode que nous proposons sont résumées dans l'algorithme suivant

Algorithm 7 Méthode de Krylov sans maillage pour les équations de type Burgers stationnaires

1. On choisit un premier terme X_0 , une tolérance tol , un entier $kmax$ et on pose $k = 0$
 2. Calculer $Y_k = \mathcal{F}(X_k)$ et sa norme de Frobenius $\|Y_k\|_F$
 3. **Si** $\|Y_k\|_F < tol$ ou $k > kmax$ **alors**
 4. Calculer l'approximation Λ_k à Λ en utilisant (3.24) :

$$\Lambda_k = S^{-1} \left(X_k - [M \quad Q_1] A^{-1} \begin{bmatrix} G \\ O_1 \end{bmatrix} \right)$$
 5. Calculer les approximations Γ_k et Θ_k à Γ et Θ en utilisant (3.24) :

$$\begin{bmatrix} \Gamma_k \\ \Theta_k \end{bmatrix} = A^{-1} \left(\begin{bmatrix} G \\ O_1 \end{bmatrix} - \begin{bmatrix} M^T \\ Q_1^T \end{bmatrix} \Lambda_k \right),$$
 6. **Stop**
 7. **Sinon**
 8. Appliquer l'algorithme 2 pour calculer la solution approchée S_k de l'équation $\mathcal{M}(Z) = -Y_k$.
 9. Calculer $X_{k+1} = X_k + S_k$
 10. $k = k + 1$ et **Aller à 2.**
 11. **Fin Si**
-

3.3.3 Exemples numériques

Nous testons la méthode exposée dans le cas $d = 2$, sur l'équation de Burgers homogène puis avec second membre non identiquement nul. Les tests sont menés en utilisant les splines plaque mince, avec précision polynomiale de degré total 2. Les centres de collocation sur $\bar{\Omega}$ (n points à l'intérieur, n' points sur le bord) sont obtenus grâce à des suites de points de Halton-Smith. Pour plus de détails quant à la génération et aux propriétés des points de Halton-Smith, l'on pourra se référer à [41].

Considérons donc l'équation

$$\begin{cases} (u(x, y) \cdot \nabla) u(x, y) - \nu \Delta u(x, y) = f(x, y), & \text{pour } (x, y) \in \Omega, \\ u(x, y) = g(x, y), & \text{pour } (x, y) \in \partial\Omega. \end{cases} \quad (3.56)$$

On estime la précision de notre méthode en calculant l'erreur relative e_r définie

par la formule

$$e_r = \frac{\sum_{x \in R \cap \Omega} \|u(x) - u_h(x)\|_2}{\sum_{x \in R \cap \Omega} \|u(x)\|_2},$$

où R est une grille uniforme de points de taille 25×25 incluse dans Ω .

Exemple 1

Dans ce premier exemple, nous résolvons l'équation (3.56) dans le domaine rectangulaire $\Omega = [-0.5, 0.5] \times [0, 1]$. La solution analytique est donnée par

$$u(x, y) = (\cos(\pi xy) + 2, \sin(\pi xy) + 2),$$

en choisissant les fonctions f et g de manière appropriée. Nous appliquons la méthode que nous avons décrite avec différents choix de nombres de points de collocation intérieurs et sur le bord. Dans les figures 3.2 et 3.3, nous représentons la solution analytique et la solution approchée sur le domaine Ω , pour $n = 60$, $n' = 160$ et $\nu = 1$.

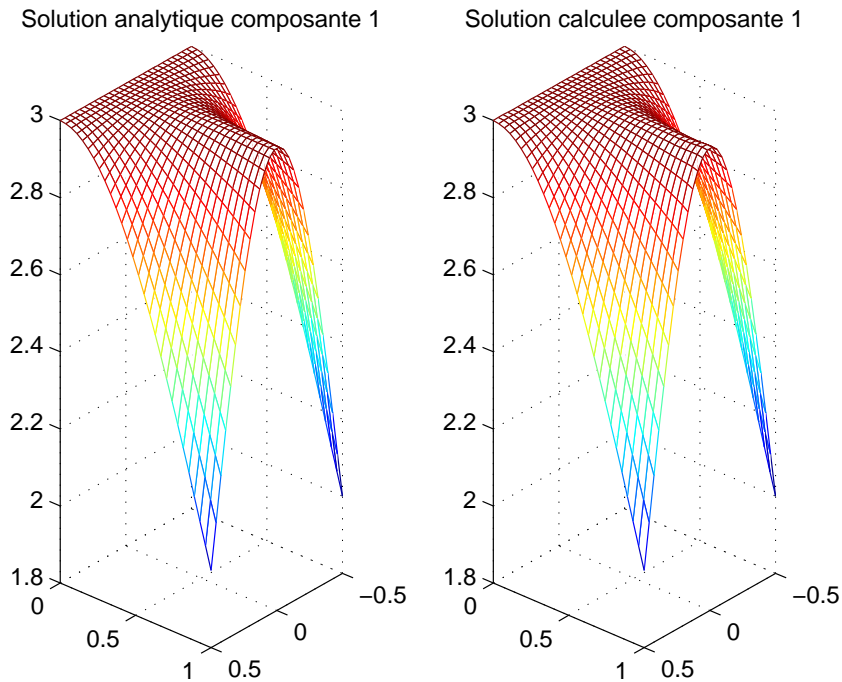


FIGURE 3.2 – Solution analytique et solution approchée : première composante.

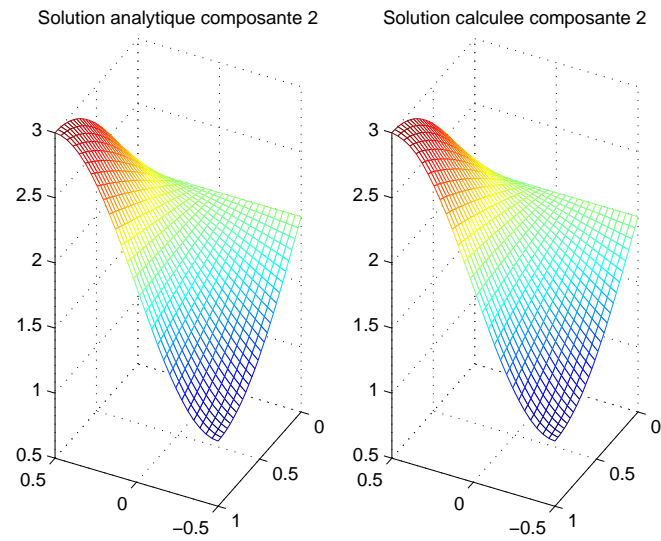


FIGURE 3.3 – Solution analytique et solution approchée : deuxième composante.

Dans la figure suivante, nous représentons l'erreur relative obtenue

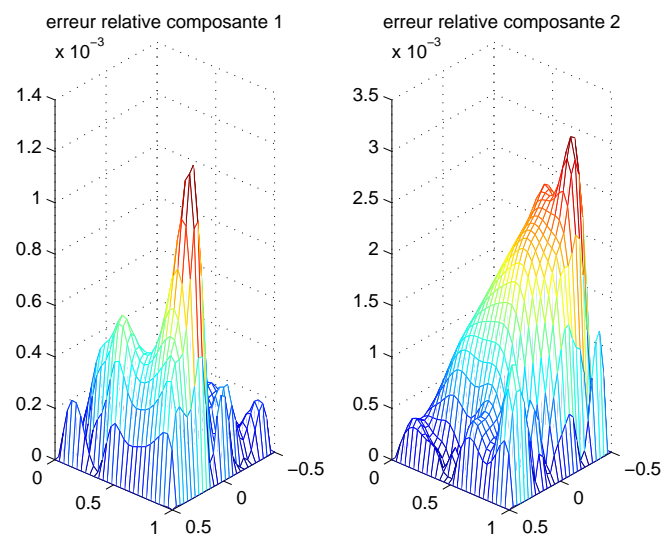


FIGURE 3.4 – Erreur relative

Dans les tableaux suivants, nous donnons les erreurs relatives pour différentes valeurs de n , n' et ν .

$N = n + n'$	n	n'	$e_r(\nu = 1)$
50	10	40	1.45e-3
80	40	40	4.52e-3
220	60	160	6.42e-4
$N = n + n'$	n	n'	$e_r(\nu = 25)$
180	20	160	7.81e-3
220	60	160	2.34e-3
660	500	160	2.36e-4

On observe que l'erreur relative décroît lorsque le nombre de points de collocation augmente. Cependant, la précision est satisfaisante dès un nombre relativement faible de points. Pour obtenir une bonne précision dans le calcul de l'inverse de la matrice A , on veille à limiter le nombre n' de points sur le bord.

Exemple 2

Dans cet exemple, on considère un ouvert connexe borné Ω qui n'est pas rectangulaire. On choisit Ω délimité par sa frontière $\partial\Omega$ qui est définie par

$$\partial\Omega = \{x = (x, y) \in \mathbb{R}^2 : x(s) = R(s) \cos(s), y(s) = R(s) \sin(s)\}, \quad (3.57)$$

où $R(s) = \sqrt{\cos(5s)^2 + \cos(2s)^2 + \cos(s)^2}$, $s \in [0, 2\pi]$.

On représente dans la figure suivante le domaine Ω et la distribution des points de Halton-Smith sur Ω et sur $\partial\Omega$.

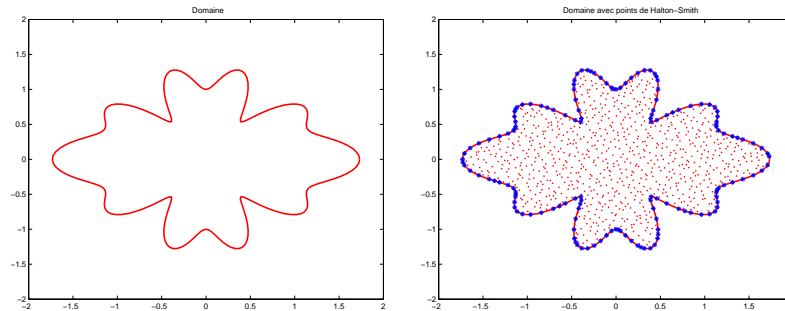


FIGURE 3.5 – Domaine Ω sans les points de Halton (à gauche), avec $n' = 150$ points sur la frontières et $n = 885$ points à l'intérieur.

Les fonctions f et g dans l'équation (3.56) sont choisies manière à ce que la solution analytique u soit donnée par

$$u(x, y) = ((x + y)e^{-x^2 - y^2}, e^{-x^2 - y^2}).$$

La figure 3.6 représente la solution analytique sous forme de champ de vecteurs et de contours pour les deux composantes de la solution analytique u et de u_h et la solution approchée par notre méthode. La résolution s'est faite pour $\nu = 0.85$, $n = 885$ points intérieurs et $n' = 150$ points sur le bord. L'erreur relative est dans ce cas $e_r = 4.23 \times 10^{-3}$. Ces figures font apparaître la qualité de l'approximation par cette méthode.

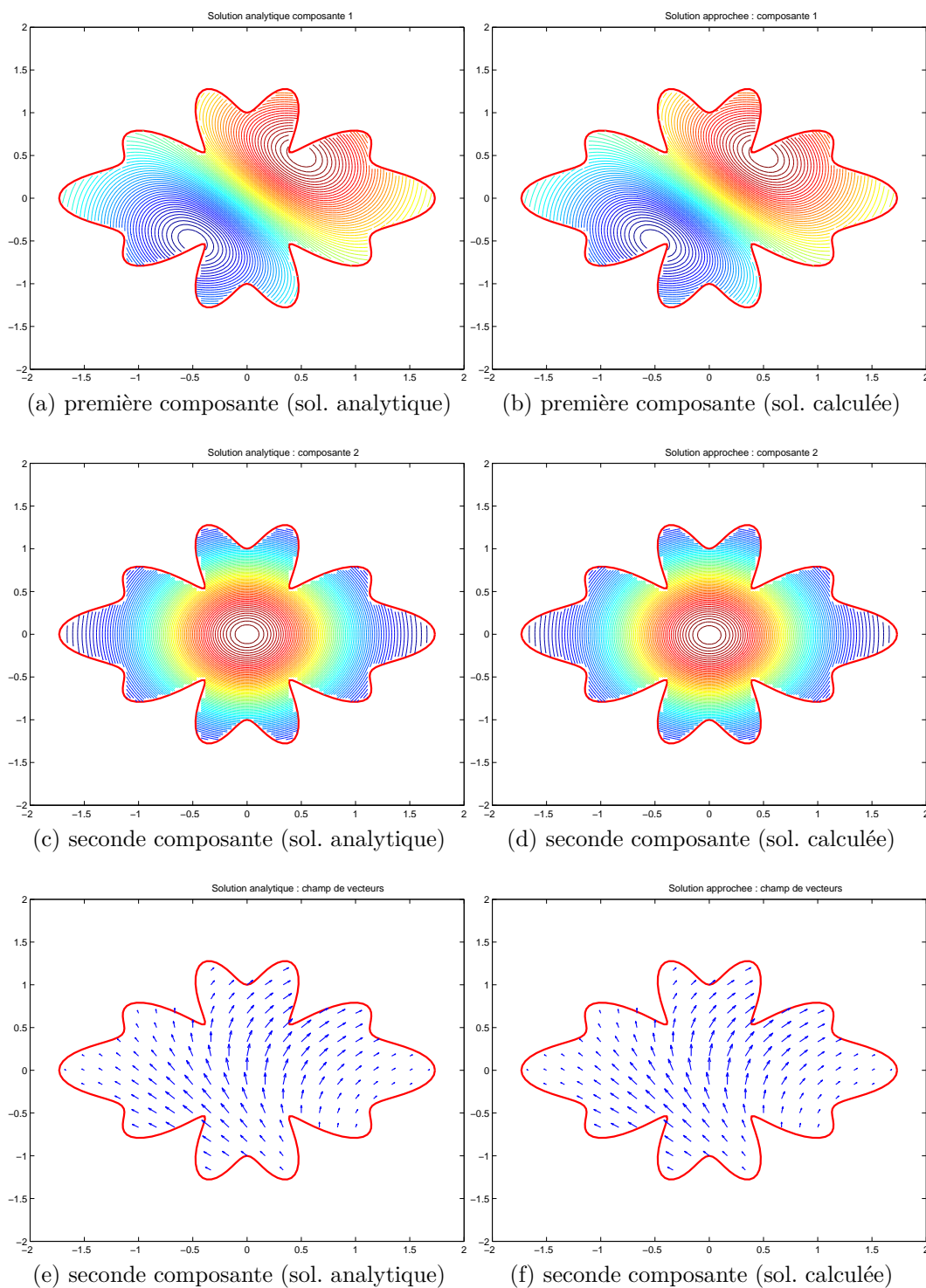


FIGURE 3.6 – Solutions analytiques (à gauche) et calculées (à droite)

Dans la figure suivante, nous représentons l'erreur relative obtenue

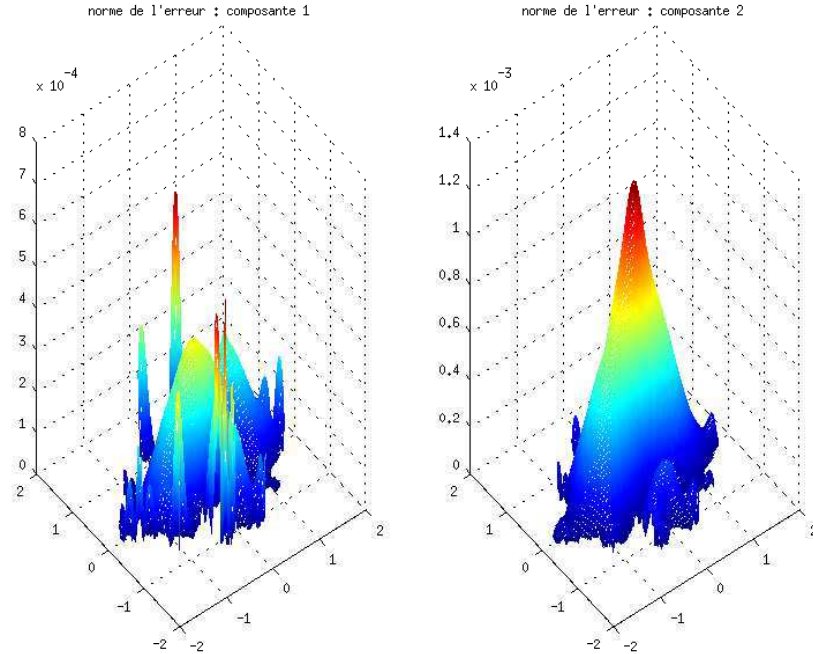


FIGURE 3.7 – Erreur relative

Dans le tableau 3.1, nous donnons les erreurs relatives obtenues pour différentes valeurs de n , n' et ν .

$N = n + n'$	n	n'	e_r pour $\nu = 0.85$	e_r pour $\nu = 10$
75	50	25	3.81e-1	3.91e-2
235	200	35	1.04e-2	2.47e-3
445	400	45	8.83e-3	8.65e-4
555	500	55	5.82e-3	8.23e-4

TABLE 3.1 – Erreurs relatives pour $\nu = 0.85$ et $\nu = 10$.

Dans ces tests numériques, le test d'arrêt choisi est $\| Y_k \|_F < 10^{-9}$. Le tableau suivant donne dans le cas $\nu = 0.85$, le nombre d'itérations de l'algorithme de Newton, ainsi que la norme du résidu.

Les résultats obtenus montrent qu'en augmentant le nombre de points, la précision est améliorée. Des essais comparatifs ont été menés en utilisant d'autres fonctions

$N = n + n'$	n	n'	nombre d'itérations	norme du résidu
75	50	25	27	8.05e-13
235	200	35	13	8.59e-11
445	400	45	14	1.06e-10
555	500	55	13	7.82e-11

TABLE 3.2 – Erreurs relatives pour $\nu = 0.85$ et $\nu = 10$.

radiales (multiquadriques) et ont donné des résultats comparables. La méthode de Newton inexacte peut nécessiter de nombreuses résolutions d'équations linéaires matricielles, dont la taille, déterminée par le nombre total de points de collocation est potentiellement importante. De plus, comme nous l'avons vu, la forme de ces équations invite à l'utilisation de la méthode GMRES globale. Dans la partie suivante, nous nous penchons sur le cas évolutif.

3.4 Méthode sans maillage pour une EDP de type Burgers évolutive

Soient Ω un domaine ouvert de l'espace réel euclidien \mathbb{R}^d , $\partial\Omega$ sa frontière et $[t_0; T]$, $T > t_0 \geq 0$ un intervalle de temps. L'équation de Burgers avec condition de Dirichlet au bord et condition initiale peut être écrite sous la forme

$$\left\{ \begin{array}{l} \frac{\partial u}{\partial t}(x, t) + (u(x, t) \cdot \nabla) u(x, t) - \nu Lu(x, t) = f(x, t), \quad x \in \Omega, \quad t \in [t_0; T] \\ u(x, t) = g(x, t), \quad x \in \partial\Omega, \quad t \in [t_0; T] \\ u(x, t_0) = u_0(x) \end{array} \right. \quad (3.58)$$

où $x = (x_1, \dots, x_d)$, $f : \Omega \times [t_0; T] \rightarrow \mathbb{R}^d$ et $g : \partial\Omega \times [t_0; T] \rightarrow \mathbb{R}^d$ sont des fonctions connues : $f(x, t) = (f_1(x, t), \dots, f_d(x, t))$, pour tout $(x, t) \in \Omega \times [t_0; T]$, $g(x, t) = (g_1(x, t), \dots, g_d(x, t))$, pour tout $(x, t) \in \partial\Omega \times [t_0; T]$.

On a

$$\begin{aligned} Lu(x, t) &= (Lu_1(x, t), \dots, Lu_d(x, t)), \\ u_0(x) &= u(x, t_0), \quad \text{pour tout } x \in \Omega, \end{aligned} \quad (3.59)$$

et

$$(u(x, t) \cdot \nabla) u(x, t) = \left(\sum_{i=1}^d u_i(x, t) \frac{\partial u_1}{\partial x_i}(x, t), \dots, \sum_{i=1}^d u_i(x, t) \frac{\partial u_d}{\partial x_i}(x, t) \right).$$

Nous supposons la frontière $\partial\Omega$ est suffisamment régulière afin d'être assuré que le problème (3.58) possède une solution u définie sur $\overline{\Omega} \times [t_0; T]$ et à valeurs dans \mathbb{R}^d .

3.4.1 Approximation sans maillage appliquée à l'équation de Burgers évolutive

Dans ce paragraphe, nous exposons une méthode sans maillage pour le calcul d'une solution numérique l'équation de Burgers visqueuse évolutive (2.1). Nous utiliserons des fonctions de type plaque mince mais toute fonction radiale suffisamment régulière conditionnellement définie positive d'ordre m pourra être utilisée. Comme dans le paragraphe précédent, l'utilisation de fonctions de type plaque mince implique la présence d'une partie polynomiale mais le formalisme de ce paragraphe pourra assez aisément être adapté à l'utilisation d'autres fonctions radiales ne nécessitant pas l'usage de polynômes. Nous reprenons une grande partie des notations du précédent paragraphe et nous ne démontrerons que les résultats pour lesquels l'introduction du temps apporte des changements.

Considérons les ensembles suivants

$$\begin{aligned} \mathcal{A}_n &= \{x_1, \dots, x_n\} \subset \Omega, \\ \mathcal{A}'_{n'} &= \{x'_1, \dots, x'_{n'}\} \subset \partial\Omega, \end{aligned} \quad (3.60)$$

où \mathcal{A}_n est un sous-ensemble fini de n points deux à deux distincts de Ω (points de collocation intérieurs) et $\mathcal{A}'_{n'}$ est un sous-ensemble fini de n' points deux à deux distincts de la frontière $\partial\Omega$ (points de collocations sur la frontière). Nous supposons que $\mathcal{A}'_{n'}$ contient un sous-ensemble $\Pi_{m-1}(\mathbb{R}^d)$ -unisolvant de points deux à deux distincts. Notons $N = n + n'$ le nombre total de points de collocation dans $\overline{\Omega}$ et définissons l'ensemble $\mathcal{A}_N = \mathcal{A}_n \cup \mathcal{A}'_{n'}$.

On note h la dispersion de \mathcal{A}_N dans $\overline{\Omega}$. Soit $u = (u_1, \dots, u_d) : \overline{\Omega} \times [t_0; T] \longrightarrow \mathbb{R}^d$ la solution analytique de l'équation (3.58) et définissons X la matrice inconnue de taille $n \times d$ donnée par

$$X(t) = \begin{bmatrix} u_1(x_1, t) & \dots & u_d(x_1, t) \\ \vdots & & \vdots \\ u_1(x_n, t) & \dots & u_d(x_n, t) \end{bmatrix}, \quad \text{pour } t \in [t_0; T]. \quad (3.61)$$

D'après le théorème 3.6, à chaque instant $t \in [t_0, T]$, il existe une unique fonction à base radiale $u_h(\cdot, t)$ qui interpole la solution $u(\cdot, t)$ sur l'ensemble \mathcal{A}_N . Cet interpolant $u_h(\cdot, t)$ peut s'écrire pour tout $x \in \overline{\Omega}$ et en chaque instant $t \in [0; T]$ comme

suit

$$u_h(\mathbf{x}, t) = \sum_{i=1}^n (\lambda_{i,1}(t), \dots, \lambda_{i,d}(t)) \phi_m(\|\mathbf{x} - \mathbf{x}_i\|) + \sum_{i=1}^{n'} (\gamma_{i,1}(t), \dots, \gamma_{i,d}(t)) \phi_m(\|\mathbf{x} - \mathbf{x}'_i\|) + \sum_{i=1}^{d_m} (\theta_{i,1}(t), \dots, \theta_{i,d}(t)) q_i(\mathbf{x}), \quad (3.62)$$

soumis aux conditions d'orthogonalité

$$\sum_{i=1}^n (\lambda_{i,1}(t), \dots, \lambda_{i,d}(t)) q_j(\mathbf{x}_i) + \sum_{i=1}^{n'} (\gamma_{i,1}, \dots, \gamma_{i,d}(t)) q_j(\mathbf{x}'_i) = 0, \quad j = 1, \dots, d_m. \quad (3.63)$$

Nous introduisons les notations

$$\Lambda(t) = \left[\lambda_{i,j}(t) \right]_{\substack{1 \leq i \leq n \\ 1 \leq j \leq d}}, \quad \Gamma(t) = \left[\gamma_{i,j}(t) \right]_{\substack{1 \leq i \leq n' \\ 1 \leq j \leq d}}, \quad \text{et} \quad \Theta(t) = \left[\theta_{i,j}(t) \right]_{\substack{1 \leq i \leq d_m \\ 1 \leq j \leq d}}. \quad (3.64)$$

Les conditions d'interpolation couplées à celles d'orthogonalité peuvent être formulées sous forme matricielle

$$u_h(\mathbf{x}, t) = \lambda_m(\mathbf{x})^T \Lambda(t) + \gamma_m(\mathbf{x})^T \Gamma(t) + \theta_m(\mathbf{x})^T \Theta(t). \quad (3.65)$$

L'unicité de l'interpolant est équivalent à celle des matrices matrices $\Lambda(t)$, $\Gamma(t)$ et $\Theta(t)$, qui sont calculées en résolvant le système linéaire non singulier suivant

$$\begin{bmatrix} K & Q \\ Q^T & O_2 \end{bmatrix} \begin{bmatrix} \Lambda(t) \\ \Gamma(t) \\ \Theta(t) \end{bmatrix} = \begin{bmatrix} X(t) \\ G(t) \\ O_1 \end{bmatrix}, \quad (3.66)$$

où O_1 et O_2 désignent les matrices nulles de tailles $d_m \times d$ et $d_m \times d_m$ respectivement et les matrices K et Q de tailles respectives $N \times N$ et $N \times d_m$ sont définies dans le paragraphe précédent.

Nous allons montrer comment obtenir une approximation de la fonction matricielle X . La fonction de type plaque mince u_h approximant la solution u sera calculée en résolvant le système linéaire non singulier (3.66).

Comme la fonction ϕ_m est conditionnellement définie positive d'ordre m sur \mathbb{R}^d , les matrices $\mathbb{A} \in \mathbb{R}^{(N+d_m) \times (N+d_m)}$ et $A \in \mathbb{R}^{(n'+d_m) \times (n'+d_m)}$ respectivement définies par

$$\mathbb{A} = \begin{bmatrix} K & Q \\ Q^T & O_2 \end{bmatrix} = \begin{bmatrix} K_1 & M & Q_1 \\ M^T & K_2 & Q_2 \\ Q_1^T & Q_2^T & O_2 \end{bmatrix} \quad \text{et} \quad A = \begin{bmatrix} K_2 & Q_2 \\ Q_2^T & O_2 \end{bmatrix}, \quad (3.67)$$

sont non singulières.

La résultat suivant est en tous points comparable à (3.7)

Proposition 3.15. *Soit $S \in \mathbb{R}^{n \times n}$ la matrice symétrique donnée par*

$$S = K_1 - [M \quad Q_1] A^{-1} \begin{bmatrix} M^T \\ Q_1^T \end{bmatrix}. \quad (3.68)$$

S est non singulière et nous avons

$$\Lambda(t) = S^{-1} \left(X(t) - [M \quad Q_1] A^{-1} \tilde{G}(t) \right) \quad \text{et} \quad \begin{bmatrix} \Gamma(t) \\ \Theta(t) \end{bmatrix} = A^{-1} \left(\tilde{G}(t) - \begin{bmatrix} M^T \\ Q_1^T \end{bmatrix} \Lambda(t) \right), \quad (3.69)$$

où $\Lambda(t)$, $\Gamma(t)$, $\Theta(t)$ sont données par (3.64)-(3.66) et $\tilde{G}(t) = \begin{bmatrix} G(t) \\ O_1 \end{bmatrix}$.

Comme dans le cas stationnaire, à chaque instant $t \in [t_0, T]$ fixé, si $X(t)$ est connu, alors on peut déterminer la solution approchée $u_h(\cdot, t)$ de l'équation (3.58). La fonction matricielle X joue donc un rôle central dans la résolution numérique de l'équation (3.58). Nous allons maintenant montrer que la fonction matricielle $X : [t_0, T] \rightarrow \mathbb{R}^{n \times d}$ est solution d'une équation différentielle ordinaire matricielle.

La proposition suivante donne une expression en fonction de X des termes de l'équation

$$\left\{ \begin{array}{l} \frac{\partial u_h}{\partial t}(x, t) + (u_h(x) \cdot \nabla) u_h(x, t) - \nu Lu_h(x, t) = f(x, t), \quad \forall (x, t) \in \mathcal{A}_n \times [t_0, T]. \end{array} \right. \quad (3.70)$$

Proposition 3.16. *La fonction u_h donnée par (3.62) peut être exprimée comme suit*

$$u_h(x, t) = a_m(x)^T X(t) + b_m(x)^T \tilde{G}(t), \quad (3.71)$$

et satisfait les relations

$$\frac{\partial u_h}{\partial t}(x, t) = a_m(x)^T X'(t) + b_m(x)^T \tilde{G}'(t), \quad (3.72)$$

$$(u_h(x, t) \cdot \nabla) u_h(x, t) = \left(a_m(x)^T X(t) + b_m(x)^T \tilde{G}(t) \right) \left([\nabla a_m(x)]^T X(t) + [\nabla b_m(x)]^T \tilde{G}(t) \right),$$

et

$$Lu_h(x, t) = [La_m(x)]^T X(t) + [Lb_m(x)]^T \tilde{G}(t).$$

Démonstration. On se réfère à la démonstration de la proposition (3.7), en remarquant juste que l'identité (3.72) n'est rien d'autre que la dérivée par rapport au temps de (3.71).

□

Considérons la fonction à valeurs matricielles définie sur $\mathbb{R}^d \times \mathbb{R}^{n \times d} \times [t_0; T]$ par

$$\begin{aligned} F_m(\mathbf{x}, Z, t) &= f(\mathbf{x}, t) + \nu \left([La_m(\mathbf{x})]^T Z + [Lb_m(\mathbf{x})]^T \tilde{G}(t) \right) \\ &\quad - \left(a_m(\mathbf{x})^T Z + b_m(\mathbf{x})^T \tilde{G}(t) \right) \left([\nabla a_m(\mathbf{x})]^T Z + [\nabla b_m(\mathbf{x})]^T \tilde{G}(t) \right), \end{aligned}$$

et la fonction $\mathcal{F}_m : \mathbb{R}^{n \times d} \times [t_0, T] \longrightarrow \mathbb{R}^{n \times d}$ donnée par

$$\mathcal{F}_m(Z, t) = \begin{bmatrix} F_m(\mathbf{x}_1, Z, t) \\ \vdots \\ F_m(\mathbf{x}_n, Z, t) \end{bmatrix}.$$

Théorème 3.17. *La fonction u_h donnée par (3.18) satisfait au schéma d'approximation*

$$\begin{cases} \frac{\partial u_h}{\partial t}(x, t) + (u_h(x) \cdot \nabla) u_h(x, t) - \nu Lu_h(x, t) = f(x, t), & \forall (x, t) \in \mathcal{A}_n \times [t_0, T], \\ u_h(x, t) = g(x, t), & \forall (x, t) \in \mathcal{A}'_{n'} \times [t_0, T]. \end{cases} \quad (3.73)$$

si et seulement si la fonction matricielle X définie par (3.14) est solution du système d'équations différentielles matricielles suivant

$$\begin{cases} X'(t) = \mathcal{F}_m(X(t), t), & \text{sur } t \in [t_0, T] \\ X(t_0) = X_0, \end{cases} \quad (3.74)$$

où X' désigne la dérivée de la fonction X et $X_0 = [u_0(\mathbf{x}_1)^T \dots u_0(\mathbf{x}_n)^T]^T$. Les matrices Λ , Γ et Θ figurant dans (3.18) vérifient la relation (3.24).

Démonstration. Pour tout $(\mathbf{x}, t) \in \Omega \times [t_0, T]$, l'expression

$$\frac{\partial u_h}{\partial t}(\mathbf{x}, t) + (u_h(\mathbf{x}, t) \cdot \nabla) u_h(\mathbf{x}, t) - \nu Lu_h(\mathbf{x}, t) = f(\mathbf{x}, t),$$

est équivalente à

$$a_m(\mathbf{x})^T X'(t) + b_m(\mathbf{x})^T \tilde{G}'(t) = F_m(\mathbf{x}, X(t), t) \quad (3.75)$$

En écrivant l'identité (3.75) pour $x = x_1, \dots, x_n$, on obtient

$$\begin{aligned} \mathcal{F}_m(X(t), t) &= \begin{bmatrix} \left(a_m(x_1)^T X'(t) + b_m(x_1)^T \tilde{G}'(t) \right) \\ \vdots \\ \left(a_m(x_n)^T X'(t) + b_m(x_n)^T \tilde{G}'(t) \right) \end{bmatrix} \\ &= \begin{bmatrix} a_m^T(x_1) \\ \vdots \\ a_m^T(x_n) \end{bmatrix} X'(t) + \begin{bmatrix} b_m^T(x_1) \\ \vdots \\ b_m^T(x_n) \end{bmatrix} \tilde{G}'(t). \end{aligned} \quad (3.76)$$

Il suffit d'appliquer la proposition (3.3.1) pour arriver à la conclusion. \square

3.4.2 Calcul des coefficients matriciels

Comme dans le chapitre 1, nous allons appliquer un schéma de Runge-Kutta implicite à l'équation (3.74) qui va transformer le problème en une équation matricielle non linéaire à laquelle nous appliquerons l'algorithme Global-GMRES.

On considère donc l'équation

$$\begin{cases} X'(t) = \mathcal{F}_m(X(t), t), \text{ sur } t \in [t_0, T] \\ X(t_0) = X_0, \end{cases} \quad (3.77)$$

où $X_0 = [u_0(x_1)^T \dots u_0(x_n)^T]^T$.

On note X_k l'approximation de la solution X de l'équation (3.82) à l'instant $t = t_k$ par la méthode que nous allons décrire. Les seuls changements proviendront du fait que les inconnues considérées sont des matrices de taille $n \times d$ et non des vecteurs comme dans le chapitre 1.

Soit $t_k = t_0 + k\delta t$, $0 \leq k \leq N$ une discrétisation de l'intervalle de temps $[t_0, T]$, où $\delta t = \frac{T - t_0}{N}$ est la taille du pas que l'on a choisi. Soit $s \geq 1$ un entier. On se reportera au premier chapitre en ce qui concerne les coefficients de Butcher de la méthode. Le schéma de Runge-Kutta à s étapes permet de passer de (X_k, t_k) à (X_{k+1}, t_{k+1}) en procédant comme suit

En notant $Y_i \in \mathbb{R}^{n \times d}$, $1 \leq i \leq s$ l'approximation de $X(t_k + c_i \delta t)$, on définit la fonction $\mathbb{F}_k : \mathbb{R}^{ns \times d} \rightarrow \mathbb{R}^{ns \times d}$,

définie par

$$\mathbb{F}(Y, t) = [\mathcal{F}(Y_1, t_k + c_1 \delta t)^T, \dots, \mathcal{F}(Y_s, t_k + c_s \delta t)^T]^T,$$

et l'on résout le système implicite

$$Y = e \otimes X_k + \delta t (\tilde{A} \otimes I_n) \mathbb{F}_k(Y, t_k), \quad (3.78)$$

où $\mathbb{Y} = \begin{bmatrix} Y_1 \\ \vdots \\ Y_s \end{bmatrix} \in \mathbb{R}^{ns \times d}$. Une fois la matrice \mathbb{Y} connue, on calcule l'approximation $X_{k+1} \in \mathbb{R}^{n \times d}$ par la formule

$$X_{k+1} = X_k + \delta t (b^T \otimes I_n) \mathbb{F}_k(\mathbb{Y}, t_k). \quad (3.79)$$

Comme nous l'avons déjà vu dans le deuxième chapitre, l'équation (3.78) est équivalente à l'équation non linéaire

$$R_k(\mathbb{Y}) = 0, \quad (3.80)$$

où $R_k : \mathbb{R}^{ns \times d} \rightarrow \mathbb{R}^{ns \times d}$ est la fonction définie par

$$R_k(\mathbb{Y}) = e \otimes X_k + \delta t (\tilde{A} \otimes I_n) \mathbb{F}_k(\mathbb{Y}) - \mathbb{Y}. \quad (3.81)$$

On applique comme dans le précédent paragraphe de ce chapitre une méthode de Newton inexacte à chaque pas (X_k, t_k) . Pour simplifier les notations, nous omettons l'indice k dans R_k et \mathbb{F}_k et l'équation (3.80) est maintenant notée

$$R(\mathbb{Y}) = 0. \quad (3.82)$$

On construit donc une suite $(\mathbb{Y}_p)_{p \in \mathbb{N}}$ d'approximations de la solution exacte \mathbb{Y}^* de

(3.82), avec $\mathbb{Y}_p = \begin{bmatrix} \mathbb{Y}_{p,1} \\ \vdots \\ \mathbb{Y}_{p,s} \end{bmatrix} \in \mathbb{R}^{ns \times d}$. À partir d'un premier terme choisi $\mathbb{Y}_0 \in \mathbb{R}^{ns \times d}$, on construit \mathbb{Y}_{p+1} par la formule

$$\mathbb{Y}_{p+1} = \mathbb{Y}_p - \mathbb{S}_p, \quad (3.83)$$

où la matrice $\mathbb{S}_p \in \mathbb{R}^{ns \times d}$ est solution de l'équation matricielle linéaire

$$R'_{\mathbb{Y}_p, t_k}(\mathbb{S}) = R(\mathbb{Y}_p), \quad (3.84)$$

où la dérivée de Fréchet $R'_{\mathbb{Y}_p, t_k}$ de R en \mathbb{Y}_p à l'instant t_k est donnée par la formule

$$R'_{\mathbb{Y}_p, t_k}(\mathbb{H}) = \delta t (\tilde{A} \otimes I_n) D_{\mathbb{Y}_p} \mathbb{F}(\mathbb{H}) - \mathbb{H}, \text{ pour tout } \mathbb{H} \in \mathbb{R}^{ns \times d}. \quad (3.85)$$

Quant à la dérivée de Fréchet $\mathbb{F}'_{\mathbb{Y}_p, t_k}$ de \mathbb{F} en \mathbb{Y}_p à l'instant t_k , on a

$$\mathbb{F}'_{\mathbb{Y}_p, t_k}(\mathbb{H}) = \begin{bmatrix} \mathcal{F}'_{(\mathbb{Y}_{p,1}; t_k + c_1 \delta t)}(H_1) \\ \vdots \\ \mathcal{F}'_{(\mathbb{Y}_{p,s}; t_k + c_s \delta t)}(H_s) \end{bmatrix}, \text{ pour } \mathbb{H} = \begin{bmatrix} H_1 \\ \vdots \\ H_s \end{bmatrix} \in \mathbb{R}^{ns \times d}. \quad (3.86)$$

Comme dans le paragraphe précédent, le calcul de l'incrément \mathbb{S}_p se fera par l'algorithme de Global-GMRES. Nous appliquerons en fait une méthode de type quasi-Newton inexacte car c'est une approximation de la dérivée de Fréchet de \mathbb{F} qui sera utilisée et l'équation sera résolue numériquement. En effet, dans le but de gagner en temps de calcul, nous approximations chaque dérivée intermédiaire $\mathcal{F}'_{(\mathbb{Y}_p, i; t_k + c_i \delta t)}(H_i)$, pour $1 \leq i \leq s$ de \mathcal{F} par $\mathcal{F}'_{(X_k, \hat{t}_k)}(H_i)$, $1 \leq i \leq s$, où \hat{t}_k est un temps intermédiaire choisi entre t_k et t_{k+1} . On obtient facilement que

$$\begin{aligned} \mathcal{F}'_{(X_k; \hat{t}_k)}(H_i) &= - \begin{bmatrix} a_m^T(x_1) H_i \left(\nabla a_m^T(x_1) X_k + \nabla b_m^T(x_1) \right) \\ \vdots \\ a_m^T(x_n) H_i \left(\nabla a_m^T(x_n) X_k + \nabla b_m^T(x_n) \right) \end{bmatrix} \\ &- \begin{bmatrix} \left(a_m^T(x_1) X_k + b_m^T(x_1) \begin{bmatrix} G(\hat{t}_k) \\ 0_1 \end{bmatrix} \right) \nabla a_m^T(x_1) - \nu L a_m^T(x_1) \\ \vdots \\ \left(a_m^T(x_n) X_k + b_m^T(x_n) \begin{bmatrix} G(\hat{t}_k) \\ 0_1 \end{bmatrix} \right) \nabla a_m^T(x_n) - \nu L a_m^T(x_n) \end{bmatrix} \cdot H_i, \end{bmatrix} \end{aligned} \quad (3.87)$$

expression qui peut se reformuler de la façon suivante

$$\mathcal{F}'_{(X_k; \hat{t}_k)}(H_i) = -B_{i,k} - W^T H_i, \quad (3.88)$$

où

$$B_{i,k} = \begin{bmatrix} u_1^T H_i v_1^T(X_k) \\ \vdots \\ u_n^T H_i v_n^T(X_k) \end{bmatrix},$$

$u_i = a_m(x_i) = e_i \in \mathbb{R}^n$, $v_i(X_k) = X_k^T \nabla a_m(x_i) + \nabla b_m(x_i) \in \mathbb{R}^{d \times d}$, et $W^T = [w_1, \dots, w_s]$, avec

$$\begin{aligned} w_i &= \nabla a_m(x_i) \left(X_k^T a_m(x_i) + [G(\hat{t}_k)^T O_1^T] b_m(x_i) \right) - \nu L a_m(x_i) \\ &= \nabla a_m(x_i) X_k^T e_i - \nu L a_m(x_i) \in \mathbb{R}^n. \end{aligned} \quad (3.89)$$

Donc, la dérivée de Fréchet $R'_{\mathbb{Y}_p}$ donnée en (3.85) peut être approximée par l'opérateur linéaire

$$\widehat{D}_{\mathbb{Y}_p} : \mathbb{R}^{ns \times d} \longrightarrow \mathbb{R}^{ns \times d}$$

défini par

$$\widehat{D}_{\mathbb{Y}_p}(\mathbb{H}) = -\delta t (\tilde{A} \otimes I_n) \tilde{B}_{s,k} - \delta t (\tilde{A} \otimes I_n) (I_s \otimes W^T) \mathbb{H} - \mathbb{H}, \quad (3.90)$$

où

$$\tilde{B}_{s,k} = \begin{bmatrix} B_{1,k} \\ \vdots \\ B_{s,k} \end{bmatrix}.$$

En appliquant l'identité $(\tilde{A} \otimes I_n)(I_s \otimes W^T) = \tilde{A} \otimes W^T$, nous avons

$$\widehat{D}_{\mathbb{Y}_p}(\mathbb{H}) = -\delta t(\tilde{A} \otimes I_n)\tilde{B}_{s,k} - \delta t(\tilde{A} \otimes W^T)\mathbb{H} - \mathbb{H}. \quad (3.91)$$

Donc, au lieu de résoudre l'équation linéaire matricielle (3.84), nous résoudrons l'équation approchée

$$\widehat{D}_{\mathbb{Y}_p}(\mathbb{S}) = R(\mathbb{Y}_p). \quad (3.92)$$

Comme dans le cas stationnaire, la résolution numérique de l'équation (3.92) se fera à l'aide de l'algorithme Global-GMRES.

3.4.3 Exemples numériques

Dans cette partie, nous donnons quelques exemples numériques pour illustrer notre approche. Pour chaque exemple, le domaine choisi est l'ouvert $\Omega \in \mathbb{R}^2$ délimité par la courbe paramétrée $\partial\Omega$ définie par

$$\partial\Omega = \{x = (x, y) \in \mathbb{R}^2 : x(s) = R(s) \cos(s), y(s) = R(s) \sin(s)\},$$

où

$$R(s) = \frac{1}{2} \sqrt{\cos(5s)^2 + \cos(2s)^2 + \cos(s)^2}, \quad s \in [0, 2\pi].$$

Comme dans le cas stationnaire, la solution approchée u_h a été calculée en utilisant des points de Halton-Smith [36, 41]. L'interpolation a été faite par des fonctions splines de type plaque mince (TPS). En parallèle, nous avons mené les mêmes expériences avec des fonctions multiquadriques qui ont donné des résultats comparables. Chaque exemple traité est choisi de telle manière que la solution analytique u est connue afin de pouvoir estimer l'erreur commise. Cette erreur est estimée en calculant l'erreur relative

$$e_r = \frac{\sum_{x \in R \cap \Omega} \|u(x, T) - u_h(x, T)\|_2}{\sum_{x \in R \cap \Omega} \|u(x, T)\|_2},$$

où R est une grille rectangulaire uniforme de taille 25×25 contenue dans Ω .

3.4.4 Exemple 1

Pour le premier exemple, nous considérons l'équation de Burgers en deux dimensions

$$\begin{cases} \frac{\partial u}{\partial t} - \nu \Delta u + (u \cdot \nabla) u = 0, & \text{sur } \Omega \times [0; 1] \\ u = g, & \text{sur } \partial\Omega \times [0; 1] \\ u(\cdot, 0) = u_0, & \text{sur } \Omega. \end{cases} \quad (3.93)$$

où g est choisie égale à la restriction au bord du domaine de la fonction u obtenue par la transformation de Hopf-Cole [37], donnée par la formule

$$u(x, y, t) = \left(\frac{3}{4} - \frac{1}{4} \cdot \frac{1}{1 + e^{\frac{-4x+4y-t}{32\nu}}} ; \frac{3}{4} + \frac{1}{4} \cdot \frac{1}{1 + e^{\frac{-4x+4y-t}{32\nu}}} \right). \quad (3.94)$$

Nous avons essayé plusieurs tableaux de Butcher pour les coefficients de la méthode de Runge-Kutta implicite. De bons résultats ont été obtenus par plusieurs d'entre elles (sans différences significatives). Nous reportons ici ceux qui ont été obtenus par la méthode d'Euler implicite et la méthode SDIRK (Singly Diagonally Implicit Runge Kutta) à deux étapes dont le tableau est donné dans le chapitre 2.

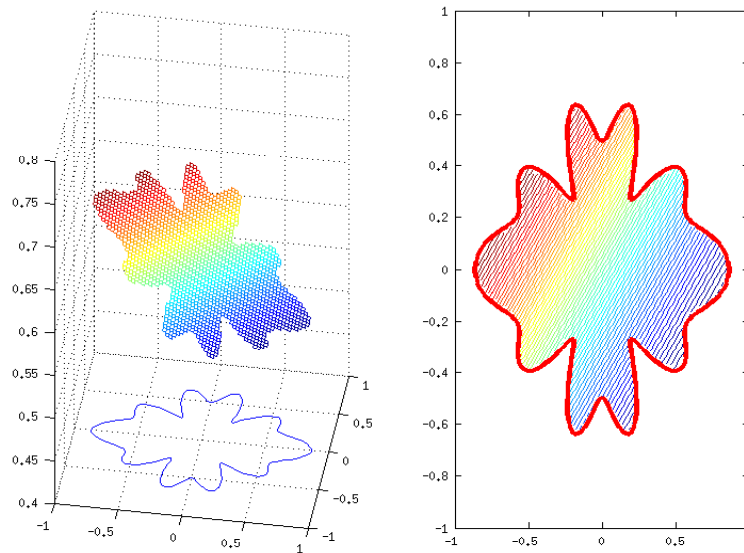


FIGURE 3.8 – Solution analytique (première composante)

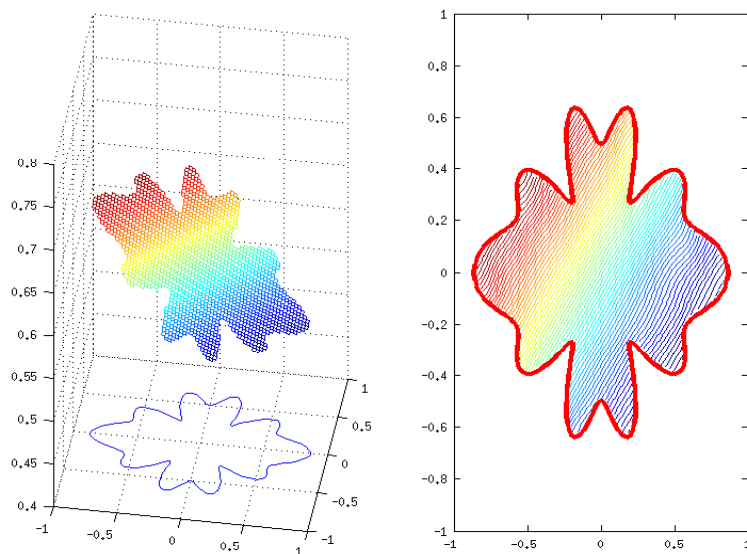


FIGURE 3.9 – Solution approchée (première composante)

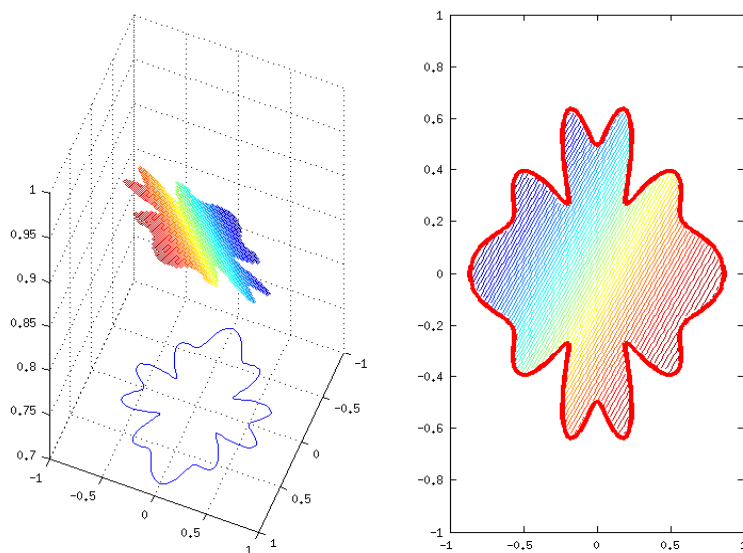


FIGURE 3.10 – Solution analytique (seconde composante)

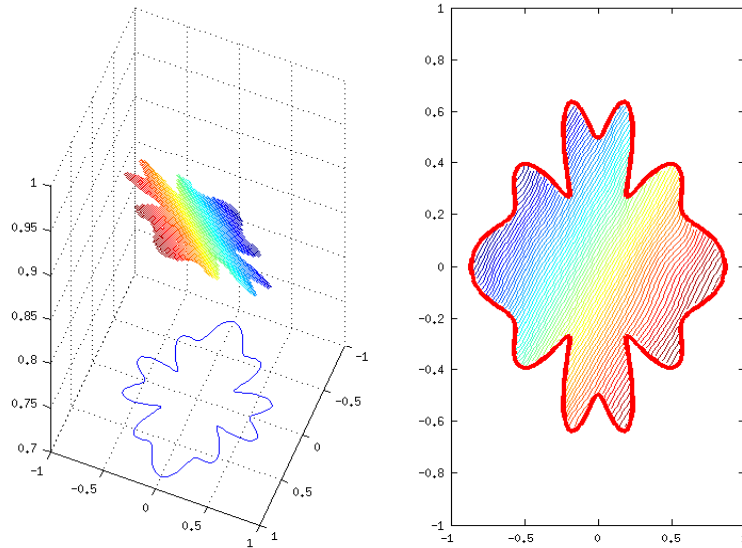


FIGURE 3.11 – Solution approchée (seconde composante)

Dans les figures 3.8 à 3.11, nous représentons les deux composantes des solutions approchées et analytiques sous forme de contours à l'instant $t = 1$, avec un pas $\delta t = 0.01$. L'interpolation a été faite en $n = 50$ points intérieurs et $n' = 70$ points sur la frontière. Le paramètre ν a été choisi égal à 0.1. Comme on peut l'observer sur les figures, les résultats sont satisfaisants (erreur relative $e_r = 9.03 \times 10^{-3}$), bien que le nombre de points de collocation soit relativement peu important.

Dans le tableau 3.3, nous donnons quelques erreurs relatives obtenues à l'instant $T = 1$, pour des valeurs différentes du nombre de points de collocation et du paramètre ν . Le pas de temps δt est fixé à 0.01.

n	n'	$e_r(\nu = 1)$	$e_r(\nu = 0.1)$	$e_r(\nu = 0.05)$
50	70	3.85×10^{-4}	9.03×10^{-3}	5.60×10^{-2}
100	70	2.11×10^{-5}	2.85×10^{-4}	3.03×10^{-3}
100	150	1.32×10^{-5}	2.39×10^{-4}	1.41×10^{-3}

TABLE 3.3 – Erreur relative e_r pour différentes valeurs de ν

Les résultats consignés dans le tableau 3.3 montre qu'une bonne précision peut être obtenue à partir d'un nombre modéré de points. Lorsque le paramètre ν décroît, le problème est dominé par la convection et nous notons une perte de précision :

le nombre de points doit être augmenté. Dans le cas extrême où ν est nul, notre méthode ne donne plus satisfaction pour l'équation hyperbolique obtenue.

Dans la figure suivante, nous représentons l'erreur relative pour $\nu = 1$, $n = 100$ points intérieurs, $n' = 70$ points au bord en utilisant le schéma d'Euler implicite avec un pas de temps $\delta = 0.01$.

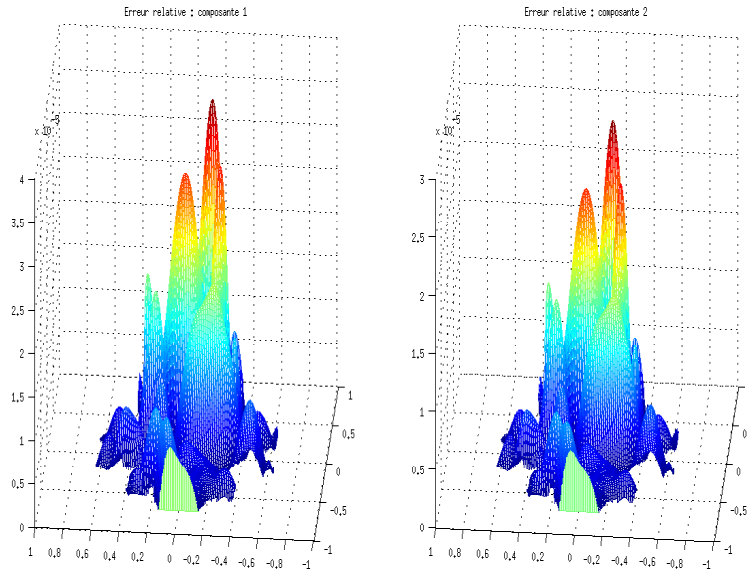


FIGURE 3.12 – Erreur absolue

3.4.5 Exemple 2

On considère maintenant l'équation de Burgers non homogène

$$\begin{cases} \frac{\partial u}{\partial t} - \nu \Delta u + (u \cdot \nabla) u = f, & \text{sur } \Omega \times [0; 1] \\ u = g, & \text{sur } \partial\Omega \times [0; 1] \\ u(., 0) = u_0, & \text{sur } \Omega. \end{cases} \quad (3.95)$$

Les fonctions f et g sont choisies de telle manière que la solution exacte de l'équation (3.95) soit donnée par

$$u(x, y, t) = [(x + y)e^{-xt^2 - y^2}; e^{-x^2 - yt^2}].$$

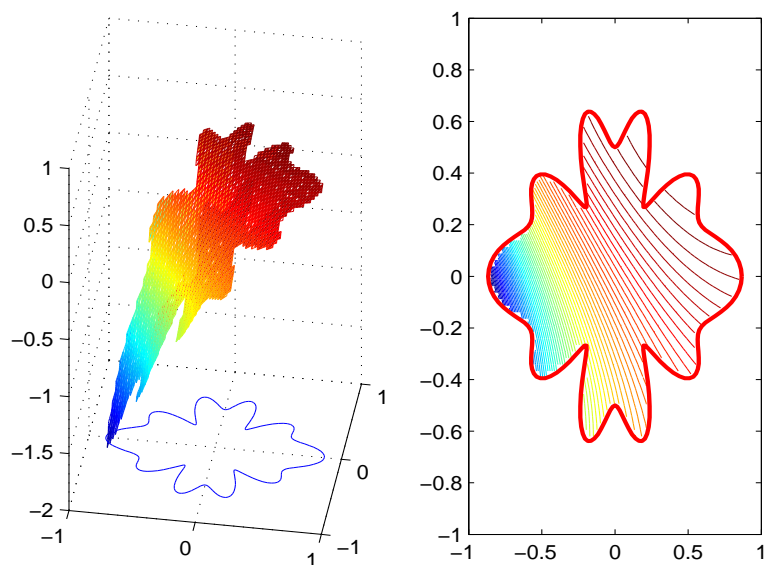


FIGURE 3.13 – Solution analytique (première composante)

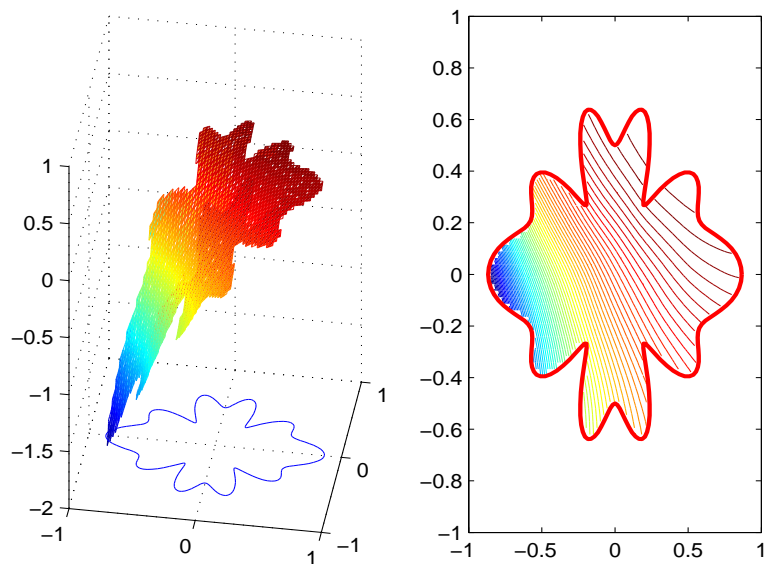


FIGURE 3.14 – Solution approchée (première composante)

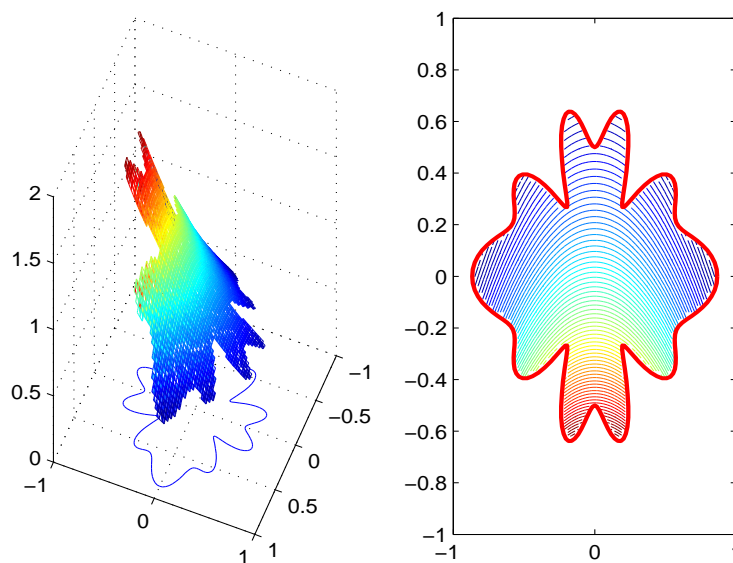


FIGURE 3.15 – Solution analytique (seconde composante)

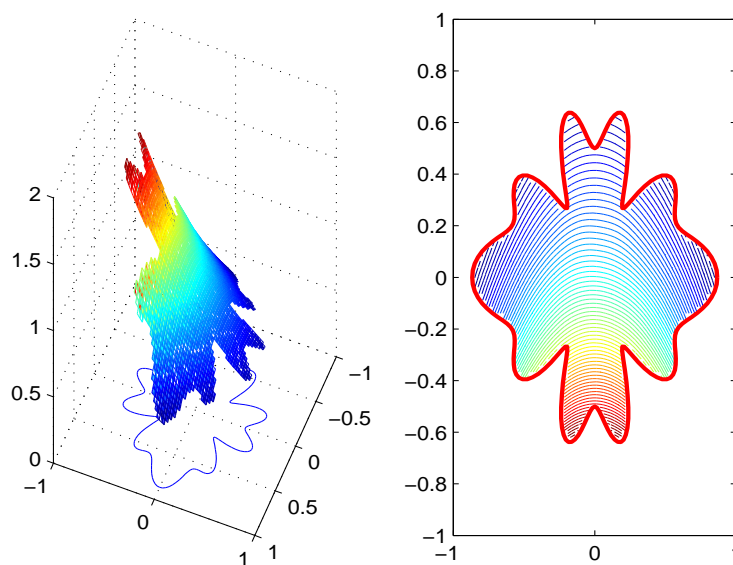


FIGURE 3.16 – Solution approchée (seconde composante)

Comme dans le cas de l'équation de Burgers homogène, nous avons constaté lors de différents essais qu'une précision satisfaisante est possible avec un nombre relativement peu important de points de collocation.

3.5 Conclusion

Dans ce chapitre, nous avons illustré l'intérêt de disposer d'outils élaborés de résolution d'équations matricielles linéaires ou non linéaires. En effet, la résolution du problème semi-discrétisé par une méthode sans maillage nécessite parfois l'utilisation de méthode d'intégrations implicites. Ces méthodes donnent lieu à des complications relativement importantes par la nature des équations linéaires matricielles qu'il est nécessaire de résoudre à chaque itération de l'algorithme de Newton. L'algorithme de GMRES global permet de s'affranchir de cette difficulté. Cependant, même si le volet algébrique de la méthode de la résolution est performant et facile à utiliser, les résultats sont décevants pour de très petites valeurs du coefficient de viscosité.

Equation matricielle de Riccati continue appliquée au contrôle des équations aux dérivées partielles.

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_∞ ou H_2 , réduction de modèle etc. On pourra se référer à [1, 29, 61] pour un exposé complet. De nombreuses méthodes numériques ont été proposées pour résoudre ces équations. On peut citer entre autres : la méthode de Newton [2, 10, 8, 60, 61], l'approche par les valeurs propres consistant à calculer les sous-espaces invariant de Lagrange d'une matrice hamiltonienne dans le cas continu [9, 61, 66, 84] ou d'un faisceau de matrices symplectiques dans le cas discret [9, 61, 66, 84]. On s'intéresse en particulier aux équations de Riccati impliquées dans les problèmes de contrôle linéaire quadratique (désigné en anglais par l'acronyme L.Q.R : Linear Quadratic Regulator) provenant de la discrétisation d'équations aux dérivées partielles avec contrôle.

Le problème du contrôle linéaire quadratique intervient dans de nombreux domaines scientifiques et industriels. On peut citer par exemple la mise au point d'une stratégie optimale de refroidissement d'un rail au sortir d'un laminoir afin d'accélérer le processus de production (le refroidissement doit être le plus rapide possible) tout en évitant des gradients de températures trop brutaux qui pourraient entraîner des altérations des propriétés mécaniques du rail ou encore la nécessité de contrôler l'écoulement de l'air ou de l'eau sur un véhicule ou un pont pour éviter la formation de phénomènes indésirables (tourbillons dans le sillage d'un avion, oscillations trop fortes d'un pont).

Ce contrôle peut être passif (on adapte la géométrie du pont ou de l'aile d'un avion) ou actif (on refroidit le rail sortant du laminoir par pulvérisation contrôlée de fluide

refroidissant. On se penche sur le contrôle actif pour lequel on exerce une action sur le système. Les phénomènes étudiés sont souvent régis par des équations aux dérivées partielles dépendant du temps. Nous nous limitons au cadre d'équations aux dérivées partielles linéaires. En ce qui concerne les techniques utilisées en contrôle non linéaire, on pourra se référer à [81, 59].

La première étape consiste en la semi-discrétisation de l'équation aux dérivées partielles par éléments finis, différences finies ou volumes finis. On aboutit à un système d'équations différentielles linéaires de la forme

$$\dot{x}(t) = Ax(t) + Bu(t) \quad (4.1)$$

$$y(t) = Cx(t); x(0) = x_0, \quad (4.2)$$

où les matrices $A \in \mathbb{R}^{n \times n}$ et $B \in \mathbb{R}^{n \times r}$ proviennent de la discrétisation de l'équation aux dérivées partielles, le vecteur $x(t) \in \mathbb{R}^n$ est appelé vecteur d'état, le vecteur $u(t) \in \mathbb{R}^r$ désigne le vecteur de contrôle, la matrice $C \in \mathbb{R}^{r \times n}$ pondère le vecteur d'état et $y(t) \in \mathbb{R}^r$ désigne le vecteur de retour ou réponse (bien souvent, on utilise le mot anglais *feedback* dans la littérature en langue française).

Les tailles des matrices A et B , qui dépendent du nombre n de sommets ou de points utilisés pour la semi-discrétisation de l'équation aux dérivées partielles considérée, peuvent être très grandes.

En fonction des objectifs recherchés, on cherche à déterminer une solution u qui minimise la fonctionnelle

$$J(x_0, u) = \int_0^{+\infty} [x^T(t)Qx(t) + u^T(t)Ru(t)]dt, \quad (4.3)$$

où les matrices symétriques $Q \in \mathbb{R}^{n \times n}$ et $R \in \mathbb{R}^{r \times r}$ qui pondèrent l'état et le contrôle sont choisies semi-définies positives et définies positives respectivement en fonction des objectifs recherchés. On pourra se reporter à l'ouvrage [54] pour plus de détails quant au choix de ces matrices.

On démontre que la solution u au problème (4.3) [38] existe et est donnée par

$$u(t) = -RB^T Xx(t), \quad (4.4)$$

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

$$A^T X + XA - XBR^{-1}B^T X + C^T QC = 0 \quad (4.5)$$

Les théorèmes portant sur les conditions d'existence et les propriétés d'une telle solution sur lesquelles nous reviendrons plus loin sont énoncés et démontrés dans [61] entre autres.

4.2 Méthode de Newton

Nous nous attachons dans ce paragraphe suivant à résoudre l'équation de Riccati continue (4.5). Remarquons que comme les matrices symétriques Q et R sont positives semi-définie et définie respectivement, nous pouvons, moyennant une réécriture des matrices A et B en tenant compte des matrices Q et R , considérer l'équation à résoudre sous la forme

$$A^T X + X A - X B B^T X + C^T C = 0, \quad (4.6)$$

où $A, X \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times r}$ et $C \in \mathbb{R}^{r \times n}$. Nous supposons que les matrices B et C sont de rang maximal r et $r \ll n$. Sous les conditions énoncées dans [88] : 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 $J(x_0, u)$ est minimisée par

$$u(t) = -B^T X x(t),$$

où $X \in \mathbb{R}^{n \times n}$ est l'unique solution symétrique semi-définie stabilisante ($\operatorname{Re}(\lambda(A - BB^T X)) < 0$) de l'équation (4.6).

Pour les problèmes de petite dimension, les méthodes exactes de Schur [63] et de la fonction signe matricielle [9, 26] sont employées mais elles ne sont pas envisageables lorsque n est grand, ce qui se produit par exemple dans les problèmes de contrôle linéaire issus de la discrétisation par éléments finis ou différences finies d'équations aux dérivées partielles. Dans ce cas, des méthodes numériques utilisant des projections sur sous-espaces de Krylov ont été proposées ces dernières années [45, 49, 50, 51]. Ces méthodes génèrent des approximations de petit rang de la solution de l'équation (4.6). Si elles présentent l'avantage de converger rapidement, ces méthodes ne permettent pas de s'assurer que la solution approchée soit stabilisante comme est censée l'être la solution exacte. La méthode de Newton et ses variantes sont aussi largement utilisées. On pourra se référer à [10, 8, 60, 61] entre autres.

Dans ce chapitre, nous allons combiner ces deux approches : la méthode de Newton qui à chaque étape nécessitera la résolution d'une équation de Lyapunov -cas particulier de l'équation de Sylvester- qui sera résolue numériquement par la méthode bloc Arnoldi avec préconditionnement de type ADI. Nous donnerons des résultats de majoration de l'erreur et de convergence. Nous veillerons aussi à limiter le temps de calcul et la mémoire utilisée. Enfin, des tests numériques établiront la performance de notre méthode tant sur le plan de la rapidité qu'en termes de précision.

Posons $\mathcal{R}(X) = A^T X + X A - X B B^T X + C^T C$. Nous appliquons la méthode de Newton pour résoudre l'équation

$$\mathcal{R}(X) = 0.$$

La dérivée de Fréchet de l'application non linéaire \mathcal{R} en X_k est donnée par

$$\mathcal{R}'_{X_k}(Z) = (A - B B^T X_k)^T Z + Z (A - B B^T X_k) \quad (4.7)$$

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 (4.6) définie par la relation de récurrence

$$X_{k+1} = X_k - (\mathcal{R}'_{X_k})^{-1}(\mathcal{R}(X_k)). \quad (4.8)$$

On montre que X_{k+1} peut être vu comme la solution de l'équation de Lyapunov

$$F_k^T X_{k+1} + X_{k+1} F_k + W_k W_k^T = 0 \quad (4.9)$$

où $F_k = A - B K_k$, $K_k = B^T X_k$ et $W_k = [K_k^T \ C^T]$. Remarquons qu'ici, la matrice

A est creuse alors que la matrice $F_k = A - B K_k$ ne l'est pas. L'algorithme suivant détaille les étapes de la méthode dite de Newton-Kleinman [60]

Algorithm 8 La Méthode de Newton-Kleinman pour les équations de Riccati continues

1. On choisit un premier terme $X_0 \in \mathbb{R}^{n \times n}$, on pose $K_0 = B^T X_0$ et on fixe *itermax*.
2. **Pour** $k = 1, \dots, \textit{itermax}$
 - (a) Résoudre l'équation $F_k^T X_{k+1} + X_{k+1} F_k + W_k W_k^T = 0$, où $F_k = A - B K_k$, $K_k = B^T X_k$ et $W_k = [K_k^T \ C^T]$.
 - (b) Calculer le feedback $K_{k+1} = B^T X_{k+1}$.

Fin Pour

Le théorème suivant fournit un résultat de convergence pour la méthode de Newton.

Théorème 4.1. [61] *En supposant que la matrice A est stable, et en désignant par (X_k) la suite des itérés produits par la méthode de Newton, on a*

1. *La matrice $A - B K_k$ est stable et l'équation de Lyapunov (4.9) possède une unique solution semi-définie positive.*

2. $(X_k)_{k \geq 0}$ est une suite décroissante : $X_k \geq X_{k+1} \geq 0$.
3. La convergence de la suite (X_k) est globalement quadratique vers l'unique solution stabilisante X de l'équation de Riccati (4.6).

La résolution de l'équation de Lyapunov (4.9) est l'étape essentielle de chaque itération de la méthode de Newton. Pour les petites dimensions, on peut utiliser la méthode directe de Bartels-Stewart [5]. Dans le cas où n est grand, les méthodes de projections sur des espaces de type Krylov, la méthode ADI [85] ou encore la méthode LRCF-ADI (Low Rank Cholesky Factored ADI) [10] sont les outils communément utilisés. La méthode LRCF-ADI dépend du choix de paramètres qui, s'ils ne sont pas choisis de manière optimale, font que la convergence peut être très lente. De plus, les méthodes de type Krylov demandent de nombreuses itérations pour obtenir une approximation satisfaisante. L'approche que nous adoptons, combinant un préconditionnement de type ADI suivi d'une méthode de projection à l'aide de l'algorithme de bloc-Arnoldi va nous permettre de résoudre l'équation de Lyapunov rapidement et sera économique en termes de place mémoire. Cette méthode suit en tout point les étapes décrites pour l'équation de Sylvester dans le premier chapitre de ce travail. Cependant, le caractère positif semi-défini de la solution et de ses approximations vont nous permettre d'observer des propriétés supplémentaires. C'est pourquoi nous donnerons les détails de cette méthode

4.3 Résolution de l'équation de Lyapunov

Considérons l'équation matricielle de Lyapunov

$$FX + XF^T + EE^T = 0, \quad (4.10)$$

où $F \in \mathbb{R}^{n \times n}$, $E \in \mathbb{R}^{n \times r}$ et $X \in \mathbb{R}^{n \times n}$ avec $r \ll n$. Nous supposons dans toute la suite de ce travail que la matrice F est stable, ce qui nous assure l'existence et l'unicité d'une solution.

4.3.1 Méthode ADI

La méthode ADI consiste à choisir l paramètres ADI $\mu_1, \dots, \mu_l \in \mathbb{C}_-$ et définir par récurrence la suite $(X_i^A)_{i \in \mathbb{N}}$ des approximations de la solution exacte X de l'équation (4.10) par la donnée du premier terme $X_0 = 0_{n \times n}$ et de la relation

$$X_i^A = (F - \mu_i I)(F + \mu_i I)^{-1} X_{i-1}^A (F^T - \mu_i I)(F^T + \mu_i I)^{-1} - 2\mu_i (F^T + \mu_i I)^{-1} EE^T (F + \mu_i I)^{-1}. \quad (4.11)$$

Le taux de convergence de la suite $(X_i^A)_{i \in \mathbb{N}}$ est dominé par le rayon spectral de la matrice

$$\mathcal{F}_l = \prod_{i=1}^l (F - \mu_i I)(F + \mu_i I)^{-1}, \quad (4.12)$$

où l est le nombre de paramètres. La détermination des paramètres $\mu_1, \mu_2, \dots, \mu_l$ minimisant ce rayon spectral conduit au problème de minimax

$$\{\mu_1, \mu_2, \dots, \mu_l\} = \arg \min_{(\mu_1, \mu_2, \dots, \mu_l) \in \mathbb{C}_-^l} \left(\max_{\lambda \in \sigma(F)} \frac{|(\lambda - \mu_1) \dots (\lambda - \mu_l)|}{|(\lambda + \mu_1) \dots (\lambda + \mu_l)|} \right), \quad (4.13)$$

où $\sigma(F)$ désigne le spectre de la matrice F .

L'approche classique consiste à résoudre le problème de minimax non pas sur le spectre de F que l'on ne connaît généralement pas mais sur un sous-ensemble Ω de \mathbb{C}_- le contenant. Nous utiliserons une procédure heuristique décrite dans [71, 11], qui consiste en le calcul de paramètres sous optimaux et qui est disponible sous la forme de la routine `lp-para` de Matlab. Nous ne prendrons que les parties réelles des paramètres produits par cette méthode.

4.3.2 Le préconditionnement ADI(1) de petit rang

L'équation matricielle (4.10) est équivalente l'équation de Stein symétrique

$$\mathcal{F}_l X \mathcal{F}_l^T - X + X_l^A = 0, \quad (4.14)$$

où la matrice \mathcal{F}_l de taille $n \times n$ est donnée par (4.12) et X_l^A la l -ème approximation produite par la méthode ADI. Comme $X_i^A, i = 1, 2, \dots$ est symétrique semi-définie positive, on a

$$X_i^A = Y_i^A (Y_i^A)^T \quad (4.15)$$

où le facteur Y_i^A est défini par

$$Y_i^A = [(F - \mu_i I)(F + \mu_i I)^{-1} Y_{i-1}^A \quad ; \quad \sqrt{-2\mu_i}(F + \mu_i I)^{-1} E]; \quad i \geq 2 \quad (4.16)$$

et

$$Y_1^A = \sqrt{-2\mu_1}(F + \mu_1 I)^{-1} E.$$

Dans ce travail, nous nous limitons aux cas $l = 1$ (préconditionnement de Smith) ou $l = 2$ (préconditionnement LR-ADI(2)). Remarquons que comme F est stable, pourvu que les paramètres ADI soient bien choisis réels strictement négatifs, nous avons $\rho(\mathcal{F}_l) < 1$, ce qui nous assure que la solution de l'équation de Stein (4.14) existe et est unique.

Dans la section suivante, nous expliquons comment extraire des solutions approchées de petit rang de l'équation de Stein symétrique (4.14) par une méthode de type bloc Krylov.

4.3.3 Résolution de l'équation de Stein symétrique

Soit donc à résoudre l'équation

$$R(X) = \mathcal{F}_l X \mathcal{F}_l^T - X + \tilde{E} \tilde{E}^T = 0 \quad (4.17)$$

où l'expression de $\mathcal{F}_l \in R^{n \times n}$ est donnée par (4.14) et $\tilde{E} = Y_l^A \in R^{n \times s}$ est donnée par (4.16) avec $l = 1$ ou $l = 2$.

Remarquons que comme F est stable, la matrice \mathcal{F}_l est d-stable, c'est à dire que $\rho(\mathcal{F}_l) < 1$ où $\rho(\mathcal{F}_l)$ désigne le rayon spectral de la matrice \mathcal{F}_l . Par conséquent, l'équation de Stein symétrique (4.17) (appelée stable au sens de Schur) possède une solution unique donnée par (on pourra se référer à [61])

$$X = \sum_{i=0}^{\infty} \mathcal{F}_l^i \tilde{E} \tilde{E}^T \mathcal{F}_l^{iT}. \quad (4.18)$$

On applique l'algorithme d'Arnoldi par blocs à la paire de matrices $(\mathcal{F}_l, \tilde{E})$ et obtenons une base orthonormale \mathcal{V}_m et une matrice de Hessenberg par blocs \mathcal{H}_m . Nous considérons ensuite les approximations de la solution de (4.17) sous la forme

$$X_m = \mathcal{V}_m Z_m \mathcal{V}_m^T,$$

où Z_m est solution de l'équation de Stein symétrique de dimension réduite

$$\mathcal{H}_m Z_m \mathcal{H}_m^T - Z_m + \tilde{E}_m \tilde{E}_m^T = 0. \quad (4.19)$$

avec $\tilde{E}_m = \mathcal{V}_m^T \tilde{E}$. Nous devons nous assurer que l'équation de Stein réduite (4.19) possède une unique solution. C'est le cas si l'on suppose que $\|\mathcal{F}_l\|_2 < 1$. En effet, cela implique que $\|\mathcal{H}_m\|_2 < 1$ et $\rho(\mathcal{H}_m) < 1$, ce qui nous assure de l'existence d'une solution unique symétrique semi-définie positive de l'équation (4.19).

Nous donnons maintenant un résultat de perturbation ainsi qu'une expression de la norme du résidu au pas m de notre algorithme.

Théorème 4.2. [49] *Soit X_m l'approximation de faible rang de la solution exacte X de l'équation de Stein symétrique (4.17) obtenue au pas m par l'algorithme d'Arnoldi par blocs. On a alors*

$$(\mathcal{F}_l - \mathcal{F}_{l,m}) X_m (\mathcal{F}_l - \mathcal{F}_{l,m})^T - X_m + \tilde{E} \tilde{E}^T = 0 \quad (4.20)$$

et

$$\|R(X_m)\|_F^2 = 2 \|\mathcal{H}_m Z_m E_m H_{m+1,m}^T\|_F^2 + \|H_{m+1,m} E_m^T Z_m E_m H_{m+1,m}^T\|_F^2 \quad (4.21)$$

où E_m est la matrice de taille $ms \times s$ constituée des s dernières colonnes de la matrice identité I_{ms} et $\mathcal{F}_{l,m} = V_{m+1} H_{m+1,m} V_m^T$.

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 4.3. *Au terme de m pas de l'algorithme d'Arnoldi par blocs et sous l'hypothèse*

$$\| \mathcal{F}_l \|_2 < 1,$$

nous avons

$$\| X - X_m \|_2 \leq \| Z_m \|_2 \| H_{m+1,m} \|_2 \leq \frac{2 \| \mathcal{F}_l \|_2 + \| H_{m+1,m} \|_2}{1 - \| \mathcal{F}_l \|_2^2},$$

où Z_m est la solution du problème réduit (4.19).

Démonstration. En soustrayant (4.20) de (4.17), on voit que l'erreur $X - X_m$ est l'unique solution de l'équation de Stein symétrique

$$\mathcal{F}_l (X - X_m) \mathcal{F}_l^T - (X - X_m) = -\mathcal{F}_l X_m \mathcal{F}_{l,m}^T - \mathcal{F}_{l,m} X_m \mathcal{F}_l^T + \mathcal{F}_{l,m} X_m \mathcal{F}_{l,m}^T, \quad (4.22)$$

où $\mathcal{F}_{l,m} = V_{m+1} H_{m+1,m} V_m^T$.

Comme $\rho(\mathcal{F}_l) < 1$, l'unique solution de l'équation (4.22) peut s'écrire

$$X - X_m = \sum_{i=0}^{\infty} \mathcal{F}_l^i [\mathcal{F}_l X_m \mathcal{F}_{l,m}^T + \mathcal{F}_{l,m} X_m \mathcal{F}_l^T - \mathcal{F}_{l,m} X_m \mathcal{F}_{l,m}^T] \mathcal{F}_l^{iT}. \quad (4.23)$$

Par conséquent, on a

$$\| X - X_m \|_2 \leq (2 \| \mathcal{F}_l X_m \mathcal{F}_{l,m}^T \|_2 + \| \mathcal{F}_{l,m} X_m \mathcal{F}_{l,m}^T \|_2) \sum_{i=0}^{\infty} \| \mathcal{F}_l \|_2^{2i}. \quad (4.24)$$

D'après la définition de $\mathcal{F}_{l,m}$ et le fait que $X_m = \mathcal{V}_m Z_m \mathcal{V}_m^T$, nous obtenons

$$\| X_m \mathcal{F}_{l,m}^T \|_2 = \| \mathcal{V}_m Z_m \mathcal{V}_m^T \mathcal{F}_{l,m}^T \|_2 \quad (4.25)$$

$$= \| Z_m \mathcal{V}_m^T \mathcal{F}_{l,m}^T \|_2 \quad (4.26)$$

$$\leq \| Z_m \mathcal{V}_m^T \|_2 \| \mathcal{F}_{l,m}^T \|_2 \quad (4.27)$$

$$\leq \| Z_m \|_2 \| H_{m+1,m} \|_2. \quad (4.28)$$

En utilisant l'inégalité (4.24), nous arrivons au résultat attendu. \square

Notons que dans le cas où l'on peut calculer la norme spectrale de la matrice \mathcal{F}_l et si celle-ci est strictement inférieure à 1, alors la majoration donnée par le théorème (4.3) est calculable et pourra être utilisée comme test d'arrêt pour notre algorithme. Remarquons aussi que cette majoration implique la convergence de l'algorithme en un nombre fini d'itérations ($m \leq n$).

Pour illustrer le résultat du théorème (4.3), nous donnons un exemple numérique avec pour matrice F la matrice "pde225" de la collection Matrix Market¹ avec

1. <http://math.nist.gov/MatrixMarket/>

$r = 4$. Dans la figure 4.1, les deux courbes représentent la norme de l'erreur et la borne donnée par le théorème (4.3), en fonction du nombre d'itérations m .

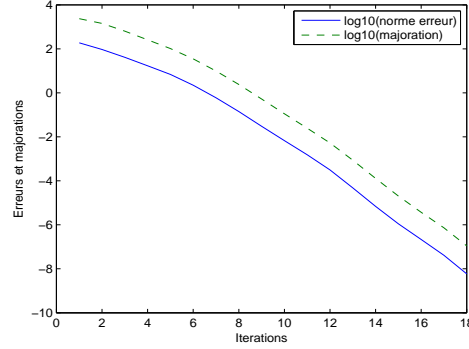


FIGURE 4.1 – norme de l'erreur et sa majoration données par le théorème (4.3)

Si de plus, l'on suppose que la matrice F est diagonalisable, on a le résultat suivant

Théorème 4.4. *Supposons la matrice F diagonalisable et notons $F = UDU^{-1}$ sa décomposition spectrale. Alors, au pas m de l'algorithme d'Arnoldi par blocs, on a*

$$\|X - X_m\|_2 \leq \kappa_2(U) \|Z_m\|_2 \|H_{m+1,m}\|_2 \frac{2 \| \mathcal{F}_l \|_2 + \| H_{m+1,m} \|_2}{1 - \rho(\mathcal{F}_l)^2}, \quad (4.29)$$

où Z_m est la solution de l'équation réduite (4.19) et $\kappa_2(U) = \|U\|_2 \|U^{-1}\|_2$.

Démonstration. Remarquons tout d'abord que comme F est stable, la matrice \mathcal{F}_l est d-stable et nous avons $\rho(\mathcal{F}_l) < 1$, ce qui nous assure l'existence et l'unicité de la solution de l'équation (4.17).

Comme $F = UDU^{-1}$, avec $D = \text{diag}(\lambda_1, \dots, \lambda_n)$, nous obtenons

$$\mathcal{F}_l = U \mathcal{D} U^{-1}, \quad (4.30)$$

où la matrice diagonale \mathcal{D} est donnée respectivement par

$$\mathcal{D} = \text{diag}\left(\frac{\lambda_1 - \mu}{\lambda_1 + \mu}, \dots, \frac{\lambda_n - \mu}{\lambda_n + \mu}\right) \quad (4.31)$$

pour le préconditionnement de Smith ($l = 1$) et

$$\mathcal{D} = \text{diag}\left(\frac{(\lambda_1 - \mu_1)(\lambda_1 - \mu_2)}{(\lambda_1 + \mu_1)(\lambda_1 + \mu_2)}, \dots, \frac{(\lambda_n - \mu_1)(\lambda_n - \mu_2)}{(\lambda_n + \mu_1)(\lambda_n + \mu_2)}\right) \quad (4.32)$$

pour le préconditionnement LR-ADI(2).

Il s'ensuit que

$$\|\mathcal{F}_l\|_2^{2i} \leq \|U\|_2 \|U^{-1}\|_2 (\rho(\mathcal{F}_l))^{2i}; \quad i \geq 0. \quad (4.33)$$

Donc, en tenant compte de (4.24), il vient

$$\|X - X_m\|_2 \leq (2 \|\mathcal{F}_l X_m \mathcal{F}_{l,m}^T\|_2 + \|\mathcal{F}_{l,m} X_m \mathcal{F}_{l,m}^T\|_2) \|U\|_2 \|U^{-1}\|_2 \sum_{i=0}^{\infty} \rho(\mathcal{F}_l)^{2i}. \quad (4.34)$$

Nous concluons en utilisant exactement les mêmes arguments que dans la démonstration du théorème précédent. \square

Si de plus \mathcal{H}_m est diagonalisable, on a le résultat

Théorème 4.5. *Soit X_m l'approximation de petit rang de la solution exacte X de l'équation (4.17). Supposons que les matrices F et \mathcal{H}_m soient diagonalisables et notons $F = UDU^{-1}$ et $\mathcal{H}_m = U_m D_m U_m^{-1}$ leurs décompositions spectrales respectives. Alors si $\rho(\mathcal{H}_m) \leq \rho(\mathcal{F}_l)$, alors on a*

$$\|X - X_m\|_2 \leq \alpha_m \frac{\rho(\mathcal{F}_l)^{2m}}{1 - \rho(\mathcal{F}_l)}$$

où $\alpha_m = \|\tilde{E}\|_2^2 (\kappa_2(U)^2 + \kappa_2(U_m)^2)$.

Démonstration. Comme $\rho(\mathcal{H}_m) \leq \rho(\mathcal{F}_l) < 1$, les solutions exactes des équations de Stein (4.17) et (4.19) sont données par

$$X = \sum_{i=0}^{\infty} \mathcal{F}_l^i \tilde{E} \tilde{E}^T \mathcal{F}_l^{iT}. \quad (4.35)$$

et

$$Z_m = \sum_{i=0}^{\infty} \mathcal{H}_m^i \tilde{E}_m \tilde{E}_m^T \mathcal{H}_m^{iT}. \quad (4.36)$$

Par conséquent, l'erreur est donnée par

$$X - X_m = \sum_{i=0}^{\infty} (\mathcal{F}_l^i \tilde{E})(\mathcal{F}_l^i \tilde{E})^T - \sum_{i=0}^{\infty} (\mathcal{V}_m \mathcal{H}_m^i \tilde{E}_m)(\mathcal{V}_m \mathcal{H}_m^i \tilde{E}_m)^T. \quad (4.37)$$

En utilisant l'identité

$$\mathcal{F}_l^i \tilde{E} = \mathcal{V}_m \mathcal{H}_m^i \tilde{E}_m; \quad i = 0, \dots, m-1,$$

on obtient

$$X - X_m = \sum_{i=m}^{\infty} (\mathcal{F}_l^i \tilde{E})(\mathcal{F}_l^i \tilde{E})^T - \sum_{i=m}^{\infty} (\mathcal{V}_m \mathcal{H}_m^i \tilde{E}_m)(\mathcal{V}_m \mathcal{H}_m^i \tilde{E}_m)^T. \quad (4.38)$$

De plus, comme

$$\|\mathcal{F}_l^i\|_2 \leq \kappa_2(U) (\rho(\mathcal{F}_l))^i \text{ et } \|\mathcal{H}_m^i\|_2 \leq \kappa_2(U_m) (\rho(\mathcal{H}_m))^i,$$

nous obtenons

$$\|X - X_m\|_2 \leq \kappa_2(U) \|\tilde{E}\|_2^2 \frac{(\rho(\mathcal{F}_l))^{2m}}{1 - \rho(\mathcal{F}_l)^2} + \kappa_2(U_m) \|\tilde{E}_m\|_2^2 \frac{(\rho(\mathcal{H}_m))^{2m}}{1 - \rho(\mathcal{H}_m)^2} \quad (4.39)$$

On conclut en utilisant le fait que $\rho(\mathcal{H}_m) \leq \rho(\mathcal{F}_l)$. \square

La solution approchée X_m peut être écrite comme produit de deux matrices de petit rang. En effet, considérons la décomposition spectrale de la matrice symétrique semi-définie $Z_m = Q \Delta Q^T \in \mathbb{R}^{ms \times ms}$, où Δ est la matrice diagonale des valeurs propres de Z_m rangées dans l'ordre décroissant. Soit Q_l la matrice de taille $ms \times l$ constituée des l premières colonnes de Q correspondant aux l valeurs propres supérieures à un seuil fixé $dtol$. On obtient la décomposition spectrale tronquée $Z_m \approx Q_l \Delta_l Q_l^T$ où $\Delta_l = \text{diag}[\lambda_1, \dots, \lambda_l]$.

En posant $Y_m = \mathcal{V}_m Q_l \Delta_l^{1/2}$, il vient

$$X_m \approx Y_m Y_m^T.$$

Les problèmes étant issus de la discrétisation d'équations aux dérivées partielles, la dimension des matrices de coefficients de l'équation de Riccati à résoudre est potentiellement très grande et le problème de la place mémoire se pose très souvent.

4.4 Algorithme de Newton Arnoldi par blocs

L'approche basée sur l'algorithme de Newton et de l'utilisation de l'algorithme d'Arnoldi par blocs que nous avons exposée peut être résumée comme ci-dessous

Algorithm 9 La méthode de Newton bloc Arnoldi ADI

- On pose $X_0 = 0$, on fixe $dtol$ et $itermax$.
 - **Pour** $k = 1, \dots, itermax$
 - Calculer les paramètres $\mu_l^{(k)}$, ($l = 1$ ou $l = 2$) pour la matrice $F_k = A^T - K_k B^T$.
 - Calculer Z_{k+1} par la méthode d'Arnoldi par blocs avec préconditionnement ADI telle que $Z_{k+1} Z_{k+1}^T$ soit une solution approchée de (4.19).
 - On calcule le feedback $K_{k+1} = Z_{k+1} (Z_{k+1}^T B)$.
 - **Fin Pour**
-

Voyons maintenant comment il est possible de limiter le coût en mémoire et le temps de calcul.

En appliquant l'algorithme d'Arnoldi par blocs pour résoudre l'équation (4.9), nous obtenons l'équation de Stein symétrique

$$\mathcal{F}_l^{(k)} X_{k+1} (\mathcal{F}_l^{(k)})^T - X_{k+1} + \widetilde{W}_{k,l} \widetilde{W}_{k,l}^T = 0, \quad (4.40)$$

où

$$\mathcal{F}_l^{(k)} = \prod_{i=1}^l (F_k - \mu_i^{(k)} I_n) (F_k + \mu_i^{(k)} I_n)^{-1}, \quad l = 1, 2, \quad (4.41)$$

$$\widetilde{W}_{k,2} = [(F_k - \mu_2^{(k)} I) (F_k + \mu_2^{(k)} I)^{-1} \widetilde{W}_{k,1} \quad \sqrt{-2\mu_1^{(k)}} (F_k + \mu_2^{(k)} I)^{-1} W_k], \quad \text{if } l = 2,$$

et

$$\widetilde{W}_{k,1} = \sqrt{-2\mu_1^{(k)}} (F_k + \mu_1^{(k)} I)^{-1} W_k, \quad \text{si } l = 1.$$

l'application de la méthode d'Arnoldi par blocs peut se révéler très coûteuse lorsque le problème est de dimension élevée. En effet, bien que la matrice A soit creuse, cela n'est pas le cas de la matrice $F_k = A^T - K_k B^T$. Par conséquent, les produits matrice-vecteurs de la forme $(F_k + \mu_i^{(k)} I)^{-1} V$ demandés par l'algorithme d'Arnoldi par blocs peuvent poser problème. En fait, la matrice $F_k = A^T - K_k B^T$ présente la particularité d'être la somme d'une matrice creuse et d'une matrice de petit rang. On peut donc utiliser la méthode de Sherman-Morrison-Woodbury

$$(\tilde{A} - UV)^{-1} = [I_n + \tilde{A}^{-1} U (I_p - V \tilde{A}^{-1} U)^{-1} V] \tilde{A}^{-1}.$$

Le temps de calcul ainsi que la place mémoire nécessaire s'en trouvent considérablement réduits. Nous concluons par une remarque quant à la méthode de Newton qui, comme l'équation de Lyapunov n'est résolue qu'approximativement, peut être qualifiée de méthode de Newton inexacte. Si la convergence est préservée, elle perd généralement son caractère quadratique.

4.5 Exemples d'application en contrôle linéaire quadratique

4.5.1 Refroidissement optimal d'un rail sortant d'un laminoir

Nous reprenons dans ce paragraphe un exemple de contrôle linéaire quadratique exposé par P. Benner et J. Saak [12] dans lequel on décrit un problème industriel. On considère la fabrication d'un rail d'acier dans un laminoir. Le procédé de fabrication est soumis à des impératifs économiques. En effet, le rail doit être refroidi

au plus vite pour pouvoir passer le plus rapidement possible à l'étape suivante de production. Cependant, un refroidissement trop brutal ou mal réparti pourrait conduire à des altérations des propriétés mécaniques du rail. Le refroidissement est effectué par pulvérisation de liquide refroidissant sur les parois du rail par des gicleurs répartis autour du rail. Il faut donc déterminer l'action de chacun de ces gicleurs de façon à ce que le refroidissement soit le plus rapide et le mieux réparti possible dans le rail.

On note $\Gamma = \cup_i \Gamma_i$ la paroi du profil du rail, noté Ω . La température θ dans le rail obéit à une équation aux dérivées partielles avec contrôle sur le bord de la forme

$$\begin{aligned} \frac{\partial \mathbf{x}}{\partial t}(\eta, t) - \nu \Delta \mathbf{x}(\eta, t) &= 0, \quad \text{sur } \Omega \times [0, +\infty[; \eta \in \Omega \subset \mathbb{R}^2 \\ \mathbf{x}(\eta, 0) &= \mathbf{x}_0(\eta), \quad \text{sur } \Omega \\ \mathbf{x}|_{\Gamma_i}(t) &= \mathbf{x}_i(t), \quad \text{sur } [0, +\infty[\end{aligned} \tag{4.42}$$

où le paramètre ν est donné par la connaissance de paramètres physiques (conductivité et capacité thermiques, densité) de la matière première. On représente la demi-section d'un rail : sur la figure de gauche, un maillage pour éléments finis et sur la figure de droite, la frontière $\Gamma = \cup_i \Gamma_i$. Cette figure est issue de la collection de problèmes de l'Université de Freiburg, accessible à l'adresse

<http://portal.uni-freiburg.de/imteksimulation/downloads/benchmark>.

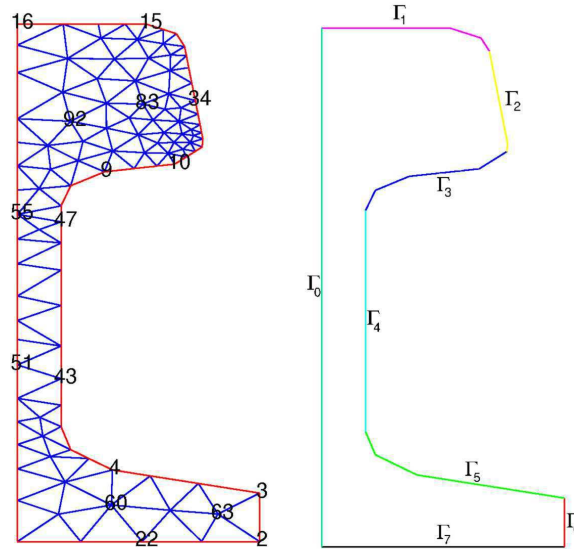
L'équation aux dérivées partielles (4.42) est discrétisée en espace par éléments finis et l'on obtient un système dynamique de la forme

$$\begin{cases} \dot{\mathbf{x}}(t) &= A \mathbf{x}(t) + \phi(\mathbf{t}) \\ \mathbf{x}(0) &= \mathbf{x}_0 \end{cases}$$

où la fonction en caractères gras \mathbf{x} est issue de la discrétisation en les noeuds du maillage de la fonction température x . Le terme ϕ provient du contrôle \mathbf{u} sur la frontière et s'écrit sous forme $\phi = B \mathbf{u}(t)$, où la matrice B décrit l'influence du contrôle sur la frontière. Comme on ne peut pas mesurer la température en tout point de Ω , on ajoute l'équation de sortie

$$z(t) = C \mathbf{x}(t),$$

où C est la matrice de sortie qui sélectionne les parties, en l'occurrence des points de la paroi Γ , en lesquels on peut mesurer la température. Pour une étude détaillée de la discrétisation d'équations aux dérivées partielles et à leur contrôle, on se réfère à [13]. L'enjeu ici est de déterminer la fonction \mathbf{u} , appelée contrôle, qui minimise une fonctionnelle de la forme



$$J(\mathbf{x}_0, \mathbf{u}) = \int_0^{+\infty} (z^T Q z + \mathbf{u}^T R \mathbf{u}) dt, \quad (4.43)$$

Nous appliquons notre algorithme de Newton-Bloc Arnoldi avec préconditionnement ADI à cette équation. Les matrices A , B et C sont issues de la page "A Semi-discretized Heat Transfer Problem for Optimal Cooling of Steel Profiles" disponible sur le site mentionné précédemment. Les matrices Q et R sont choisies égales à l'identité. Les résultats suivants donnent une comparaison en résidu et en temps de calcul entre notre méthode Newton-Krylov-ADI(2) et la méthode de Newton Low Rank Cholesky Factorized (LRCF-Newton) [10, 70]. Nous avons utilisé la fonction `lp_1rnm` du paquet Lyapack [70]. La méthode LRCF-Newton(l, k_p, k_m) demande le calcul de l paramètres ADI optimaux l obtenus après k_p itérations de l'algorithme d'Arnoldi et k_m itérations de l'algorithme d'Arnoldi inverse. Ces paramètres sont donnés par la procédure heuristique `lp_para` de Matlab qui provient de la librairie Lyapack. Le nombre d'itérations de Newton a été limité à $itermax = 20$ et pour chacune de ces applications de l'algorithme de Newton, les résolutions d'équations de Lyapunov ont été effectuées jusqu'à ce que la norme du résidu atteigne 10^{-10} ou quand leur nombre atteint $imax = 50$.

Les algorithmes ont été codés en Matlab R2011a. Comme pour les grandes tailles, le calcul de la norme des résidus est trop coûteux, le critère d'arrêt commun aux deux méthodes consiste à stopper les itérations dès que

$$\Delta_k = \|X_{k+1} - X_k\| / \|X_k\| < \epsilon = 10^{-10}.$$

Les équations de Stein de taille réduite (4.19) ont été résolues en utilisant la fonction `dlyap` de Matlab. Les équations de Lyapunov impliquées dans chacune des itérations de notre méthode étant résolues de manière approximative, la méthode de Newton est inexacte et la convergence perd son caractère quadratique.

TABLE 4.1 – Résolution de l'équation de Riccati pour $n = 1357$ et $r = 2$.

Méthode	Temps CPU	Δ_k	rang(X_k)
LRCF-Newton(10, 40, 20)	534	$4.48 \cdot 10^{-11}$	131
Newton-Block Arnoldi-ADI	346	$4.40 \cdot 10^{-12}$	131

TABLE 4.2 – Résolution de l'équation de Riccati pour $n = 5177$ et $r = 2$.

Méthode	Temps CPU	Δ_k	rang(X_k)
LRCF-Newton(10, 40, 20)	NC	NC	NC
Newton-Block Arnoldi-ADI	7392	$3,21 \cdot 10^{-12}$	147

Dans ces deux tableaux, nous avons consigné les résultats obtenus par les deux méthodes pour la résolution de l'équation de Riccati (4.6). Nous avons utilisé un paramètre ADI, calculé par la fonction `lp_para` avec $k_+ = 40$ et $k_- = 30$. Il apparaît que notre approche est performante au regard de la méthode de Newton-ADI. Dans le deuxième cas, la matrice A est de taille 5177×5177 et la méthode LRCF-Newton n'a pu être appliquée par manque de mémoire. En appliquant notre méthode aux deux cas ($n = 1357$ puis $n = 5177$), nous avons obtenu dans les deux cas des résidus de l'ordre de $\mathcal{O}(10^{-11})$. Les résultats concrets en termes de refroidissement des rails sortant du laminoir sont consultables sur le site dont sont issues les matrices ou dans [12].

4.5.2 Flux de chaleur convectif

Cet exemple décrit un modèle de flux convectif de chaleur dans un domaine donné. 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

$$\Delta \mathbf{x} - f_1(\xi) \frac{\partial \mathbf{x}}{\partial \xi_1} - f_2(\xi) \frac{\partial \mathbf{x}}{\partial \xi_2},$$

dans le domaine $\Omega = [0, 1]^2$ avec conditions de Dirichlet $\mathbf{x}(\xi, t)$, $\xi = (\xi_1, \xi_2)^T \in \Omega = [0, 1]^2$. On pose $f_1(\xi_1, \xi_2) = 10\xi_1$, $f_2(\xi_1, \xi_2) = 20\xi_1^2 \times \xi_2$.

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 $n_0 = 70$ et $r = 4$. Les coefficients des matrices $B \in \mathbb{R}^{n \times r}$ et $C \in \mathbb{R}^{r \times n}$ sont des valeurs aléatoires uniformément distribuées dans l'intervalle $[0, 1]$. Dans les tableaux 4.3 et 4.4, nous avons reporté les valeurs du temps CPU, Δ_k , le rang de l'approximation X_k et la norme relative du résidu $\|R_k\|_F / \|C^T C\|_F$, où $R_k = A^T X + XA - XBB^T X + C^T C$.

TABLE 4.3 – Résultats pour $n = 1600$ et $r = 4$.

Méthode	temps CPU	Δ_k	rang(X_k)	$\ R_k\ _F / \ C^T C\ _F$
LRCF-Newton(10, 40, 20)	84.8	$7.61 \cdot 10^{-11}$	65	$3.41 \cdot 10^{-6}$
Newton-Block Arnoldi-ADI	23.2	$3.97 \cdot 10^{-11}$	55	$2.01 \cdot 10^{-8}$

TABLE 4.4 – Résultats pour $n = 4900$ et $r = 4$.

Méthode	temps CPU	Δ_k	rang(X_k)	$\ R_k\ _F / \ C^T C\ _F$
LRCF-Newton(10, 40, 20)	493.8	$1.49 \cdot 10^{-7}$	62	$2.78 \cdot 10^{-4}$
Newton-Block Arnoldi-ADI	108.7	$2.03 \cdot 10^{-6}$	59	$1.47 \cdot 10^{-5}$

Les calculs ont été menés en suivant la même méthode que dans le cas précédent. Le paramètre ADI pour la méthode de Newton-bloc Krylov ADI(1) a été calculé avec les paramètres $k_+ = 40$ et $k_- = 20$. Ces résultats illustrent les bonnes performances de la méthode de Newton-Block Arnoldi avec préconditionnement ADI pour les problèmes de contrôle linéaire quadratique associés à des systèmes dynamiques issus de la discrétisation d'équations aux dérivées partielles.

4.6 Conclusion

Dans ce chapitre, nous avons proposé une nouvelle méthode pour la résolution numérique de l'équation de Riccati continue. Basée sur la méthode de Newton, elle se ramène à chacune des itérations à la résolution numérique d'une équation de Lyapunov. Cette dernière équation est résolue en utilisant l'algorithme d'Arnoldi par blocs avec un préconditionnement de type ADI. Des résultats de majoration de l'erreur et de convergence ont été donnés. Les tests numériques montrent l'efficacité de cette méthode.

Conclusion

Dans ce travail, nous avons donné des illustrations de l'importance de disposer d'outils algébriques performants dans la résolution numérique d'équations aux dérivées partielles et de problèmes de contrôle issus de leur semi-discrétisation.

Les contributions de ce travail dans le domaine des équations algébriques matricielles reposent en premier lieu sur une nouvelle méthode de résolution des équations de Sylvester. En effet, dans le cas où les matrices de coefficients sont creuses, de grande taille et que le second membre se décompose en un produit de deux matrices de petit rang, nous avons proposé une méthode, basée sur l'utilisation de l'algorithme de bloc-Arnoldi précédée d'un préconditionnement de type ADI. Des résultats de majoration de l'erreur ont été donnés et nous avons proposé une stratégie afin de limiter la place mémoire utilisée. Des tests numériques confirment le bon fonctionnement de cette méthode, que nous appelons LR-BA-ADI(l) (Low-rank Block-Arnoldi ADI (l), où l est le nombre de paramètres ADI choisis).

En liaison avec les problèmes de contrôle linéaire quadratique, nous avons proposé une méthode de résolution numérique de 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 de Lyapunov, à laquelle nous appliquons un algorithme dérivé du LR-BA-ADI(l). Les tests numériques donnent des résultats très encourageants.

Nous avons choisi de nous intéresser aux équations de type Burgers de la forme $\partial_t u + \mu(u \cdot \nabla)u - \nu Lu = f$, où L est un opérateur différentiel linéaire quelconque. Sur un domaine rectangulaire de \mathbb{R}^2 , pour $L = \Delta$, l'opérateur de Laplace, nous avons discrétisé l'équation en espace par différences finies et montré que le problème se ramène à un système d'équations différentielles ordinaires. Nous avons montré que l'application de la méthode de Runge-Kutta implicite conduit après linéarisation par la méthode de Newton à une équation de Stein que nous résolvons par la méthode GMRES globale. Les résultats numériques, donnés pour l'équation de la chaleur et de l'équation de Burgers sont très satisfaisants.

Nous avons ensuite voulu donner une méthode de résolution numérique de l'équation de type Burgers adaptée aux cas où le domaine est de géométrie quelconque. Nous proposons une méthode sans maillage (meshless) basée sur l'emploi

de fonctions à base radiale. Les cas stationnaire et évolutif ont été traités. Nous avons proposé un formalisme algébrique général du problème d'interpolation par des fonctions à base radiale, adapté à une dimension quelconque. Dans le cas évolutif, nous avons réécrit le formalisme algébrique de la méthode de Runge-Kutta implicite au cas matriciel. Dans les deux cas, nous avons montré que le problème se ramenait à des résolutions d'équations matricielles linéaires pour lesquelles la méthode de GMRES globale a montré toute son utilité.

Ce travail ouvre des perspectives en termes d'améliorations et d'approfondissements. On essaiera par exemple d'adapter la méthode sans maillage que nous avons proposée au cas où la viscosité ν est très petite ou même dans le cas hyperbolique ($\nu = 0$). Il serait de plus intéressant de tester la méthode sans maillage proposée en dimension 3.

Bibliographie

- [1] B.D.O Anderson and J.B. Moore. *Linear Optimal Control*. Prentice Hall, 1971.
- [2] W.F Arnold and A.J. Laub. Generalized eigenproblem algorithms and software for algebraic Riccati equations. *Proc. IEEE*, 72 :1746–1754, 1984.
- [3] A. R. Bahadir. A fully implicit finite-difference scheme for two-dimensional Burgers' equations. *Appl. Math. Comput.*, 137(1) :131–137, 2003.
- [4] A. Balzano. Mosquito : An efficient finite difference scheme for numerical simulation of 2D advection. *Int. J. Numer. Meth. Fluids*, 31 :481–496, 1999.
- [5] R.H. Bartels and G. W. Stewart. Solution of the matrix equation $AX + XB = C$ algorithm 432. *Comm. ACM*, 15 :820–826, 1972.
- [6] B. Beckermann. An error analysis for rational Galerkin projection applied to the Sylvester equation. *SIAM J. Num. Anal.*, 49 :2430–2450, 2012.
- [7] P. Benner. Factorized solution of Sylvester equations with applications in control. *Proceedings Sixteenth International Symposium on : Mathematical Theory of Network and Systems, MTNS, Leuven, Belgium*, 2004.
- [8] P. Benner and R. Byers. An exact line search method for solving generalized continuous algebraic Riccati equations. *IEEE Trans. Automat. Control*, 43 :101–107, 1998.
- [9] P. Benner and H. Faßbender. An implicitly restarted symplectic Lanczos method for the hamiltonian eigenvalue problem. *Linear Alg. Appl.*, 263 :75–111, 1997.
- [10] 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 :755–777, 2008.
- [11] P. Benner, R.C. Li, and N. Truhar. On the ADI method for Sylvester equations. *J. Comput. App. Math.*, 233 :1035–1045, 2009.
- [12] P. Benner and J. Saak. Efficient numerical solution of the LQR-problem for the heat equation. *Pamm*, 4 :648–649, 2004.

-
- [13] P. Benner, J. Saak, M. Stoll, and H.K. Weichelt. Efficient solution of large-scale saddle point systems arising in Riccati-based boundary feedback stabilization of incompressible Stokes flow. *preprint, SPP1253-130, DFG-SPP1253*, 2012.
- [14] R. Bhatia and P. Rosenthal. How and why to solve the operator equation $AX - XB = Y$. *Bull. London Math. Soc.*, 29 :1–21, 1997.
- [15] A. Bouhamidi. Weighted thin plate splines. *Anal. Appl. (Singap.)*, 3(3) :297–324, 2005.
- [16] A. Bouhamidi and K. Jbilou. Sylvester Tikhonov-regularization methods in image restoration. *Journal of Computational and Applied Mathematics*, 206(1) :86–98, 2007.
- [17] A. Bouhamidi and K. Jbilou. Meshless thin plate spline methods for the modified Helmholtz equation. *Computer Methods in Applied Mechanics and Engineering*, 197(45-48) :3733–3741, 2008.
- [18] A. Bouhamidi and K. Jbilou. Stein implicit Runge-Kutta methods with high stage order for large-scale ordinary differential equations. *Applied Numerical Mathematics*, 61 :149–159, 2011.
- [19] A. Bouhamidi and A. Le Méhauté. Multivariate interpolating (m, ℓ, s) -splines. *Adv. Comput. Math.*, 11 :287–314, 1999.
- [20] A. Bouhamidi and A. Le Méhauté. Radial basis functions under tension. *J. Approx. Theory*, 127(2) :135–154, 2004.
- [21] M.D. Buhmann. Radial functions on compact support. *Proc. Edinb. Math. Soc.*, 41(2) :33–46, 1998.
- [22] M.D. Buhmann. A new class of radial basis functions with compact support. *Math. Comput.*, 70 :307–318, 2000.
- [23] M.D. Buhmann. *Radial Basis Functions*. Cambridge University Press, Cambridge, 2003.
- [24] J.M. Burgers. Mathematical examples illustrating relations occurring in the theory of turbulent fluid motion. *Trans. Roy. Neth. Acad. Sci., Amsterdam*, 17 :1–53, 1939.
- [25] J.C. Butcher. *Numerical Methods for Ordinary Differential Equations, Second Edition*. John Wiley and Sons, Ltd, 2008.
- [26] R. Byers. Solving the algebraic Riccati equation with the matrix sign function. *Linear Algebra and its Applications*, 85 :267–279, 1987.
- [27] D. Calvetti and L. Reichel. Application of ADI iterative methods to the restoration of noisy images. *SIAM J. Matrix Anal. Appl.*, 17 :165–186, 1996.

-
- [28] J.D. Cole. On a quasi-linear parabolic equation occurring in aerodynamics. *Quart. Appl. Math.*, 9 :225–236, 1951.
- [29] B.N. Datta. *Numerical Methods for Linear Control Systems Design and Analysis*. Elsevier Academic Press, 2003.
- [30] E.D. Denman and A.N. Jr. Beavers. The matrix sign function and computations in systems. *Applied Mathematics and Computation*, 2 :63–94, 1976.
- [31] J. Duchon. Interpolation des fonctions de deux variables suivant le principe de la flexion des plaques minces. *RAIRO Analyse Numérique*, 10(12) :5–12, 1975.
- [32] J. Duchon. Splines minimizing rotation-invariant semi-norms in Sobolev spaces. *Lecture Notes in Math.*, 571 :85–100, 1977.
- [33] S.C. Eisenstat and H.F. Walker. Choosing the forcing terms in an inexact Newton Method. *SIAM J. Sci. Comput*, 17 :16–32, 1994.
- [34] A. El Guennouni, K. Jbilou, and A.J. Riquet. Block Krylov subspace methods for solving large Sylvester equations. *Numer. Alg.*, 29 :75–96, 2002.
- [35] W. Enright. Improving the efficiency of matrix operations in the numerical solution of stiff ordinary differential equations. *ACM Trans. Math. Softw.*, 4 :127–136, 1978.
- [36] G.E. Fasshauer. *Meshfree Approximation Methods with Matlab*. World Scientific, 2007.
- [37] C.A.J. Fletcher. Generating exact solutions of the two-dimensional Burgers equation. *Int. J. Numer. Meth. Fluids*, pages 213–216, 1983.
- [38] J.S. Gibson. The Riccati integral equation for optimal control problems on Hilbert spaces. *SIAM J. Cont. Optim.*, 17 :537–565, 1979.
- [39] G.H. Golub, S. Nash, and C. Van Loan. A Hessenberg-Schur method for the problem $AX + XB = C$. *IEEE Trans. Autom. Control*, 24 :909–913, 1979.
- [40] G.H. Golub and C.F. Van Loan. *Matrix Computations (Johns Hopkins Studies in Mathematical Sciences)(3rd Edition)*. The Johns Hopkins University Press, 1996.
- [41] J. H. Halton and G. B. Smith. Radical-inverse quasi-random point sequence. *Communications of the ACM*, 7(12) :701–702, 1964.
- [42] S.J. Hammarling. Numerical solution of the stable, nonnegative definite Lyapunov equation. *Ima Journal of Numerical Analysis*, 2 :303–323, 1982.
- [43] S.J. Hammarling. Numerical solution of the discrete-time, convergent non-negative definite Lyapunov equation. *Sys. Control Lett.*, 17 :137–139, 1991.
- [44] M. Heyouni. Extended Arnoldi methods for large low-rank Sylvester matrix equations. *Applied Numerical Mathematics*, 60(11) :1171–1182, 2010.

-
- [45] M. Heyouni and K. Jbilou. An extended block Arnoldi algorithm for large-scale solutions of the continuous-time algebraic Riccati equation. *Elect. Trans. Num. Anal.*, 62 :33–53, 2009.
- [46] E. Hopf. The partial differential equation $u_t + uu_x = \mu u_{xx}$. *Comm. Pure and Applied Math.*, 3 :201–230, 1950.
- [47] R.A. Horn and C.R. Johnson. *Matrix Analysis*. Cambridge University Press, 1985.
- [48] D.Y. Hu and L. Reichel. Krylov subspace methods for the Sylvester equation. *Linear Algebra and Appl.*, 174 :283–314, 1992.
- [49] I.M. Jaimoukha and E.M. Kasenally. Krylov subspace methods for solving large Lyapunov equations. *SIAM J. Matrix Anal. Appl.*, 31 :227–251, 1994.
- [50] K. Jbilou. Block Krylov subspace methods for large continuous-time algebraic Riccati equations. *Numer. Algorithms*, 34 :339–353, 2003.
- [51] K. Jbilou. An arnoldi based algorithm for large algebraic Riccati equations. *Applied Mathematics Letters*, 19 :437–444, 2006.
- [52] K. Jbilou. Low rank approximate solutions to large Sylvester matrix equations. *Appl. Math. Comp.*, 177 :365–376, 2006.
- [53] K. Jbilou, A. Messaoudi, and H. Sadok. Global FOM and GMRES algorithms for matrix equations. *Appl. Numer. Math.*, 31 :49–63, 1999.
- [54] T. Kailath. *Linear Systems*. Prentice Hall, 1980.
- [55] J. Kansa. Multiquadrics- a scattered data approximation scheme with applications to computational fluid dynamics. II. solutions to parabolic, hyperbolic partial differential equations. *Comput. Math. Appl.*, 19 :147–161, 1990.
- [56] J. Kansa. A scattered data approximation scheme with applications to computational fluid dynamics. I. Surface approximations and partial derivative estimates. *Comput. Math. Appl.*, 19 :127–145, 1990.
- [57] Y. C. Kansa, J. and Hon. Circumventing the ill-conditioning problem with multiquadric radial basis functions : Applications to elliptic partial differential equations. *Comput. Math. Appl.*, 39 :123–137, 2000.
- [58] C. T. Kelley. *Iterative Methods for Linear and Nonlinear Equations*. SIAM, Philadelphia, 1995.
- [59] H. Khalil. *Nonlinear Systems (3rd Edition)*. Prentice Hall, 2001.
- [60] D.L. Kleinman. On an iterative technique for Riccati equation computations. *IEEE Trans. Automat. Control*, 13 :114–115, 1968.
- [61] P. Lancaster and L. Rodman. *The Algebraic Riccati Equations*. Clarendon Press, Oxford, 1995.

-
- [62] P. Lancaster and M. Tismenetsky. *The Theory of Matrices, Second Edition : With Applications*. Academic Press, 1985.
- [63] A.J. Laub. A Schur method for solving algebraic Riccati equations. *IEEE Trans. Automat. control*, 24 :913–921, 1979.
- [64] A.J. Laub. *Matrix Analysis for Scientists and Engineers*. SIAM, 2004.
- [65] J.R. Li and J. White. Low rank solution of Lyapunov equations. *SIAM J. Matrix Anal. Appl.*, 24 :260–280, 2002.
- [66] V. Mehrmann. *The Autonomous Linear Quadratic Problem, Theory and Numerical Solution*. Lecture Notes in Control and Information Sciences, Vol. 63, Springer, Heidelberg, 1995.
- [67] G. Meurant. *Computer Solution of Large Linear Systems*. North Holland, Vol. 28, Studies in Mathematics and Its Applications, 1999.
- [68] C. A. Micchelli. Interpolation of scattered data : Distance matrices and conditionally positives definite functions. *Constr. Approx.*, 2 :11–22, 1986.
- [69] D.W. Peaceman and H.H. Rachford. The numerical solution of parabolic and elliptic differential equations. *J. Soc. Indust. Appl. Math.*, 3 :28–41, 1955.
- [70] T. Penzl. LYAPACK : A Matlab toolbox for large Lyapunov and Riccati equations, model reduction problems, and linear-quadratic optimal control problems. <http://www.tu-chemnitz.de/sfb393/lyapack>.
- [71] T. Penzl. A cyclic low-rank Smith method for large sparse Lyapunov equations. *SIAM J. Sci. Comput.*, 21 :1401–1418, 2000.
- [72] M. Robbé and M. Sadkane. A convergence analysis of GMRES and FOM methods for Sylvester equations. *Numerical Algorithms*, 30 :71–89, 2002.
- [73] M. Robbé and M. Sadkane. A priori error bounds on invariant subspace approximations by block Krylov subspaces. *Linear Algebra and Its Applications*, 350 :89–103, 2002.
- [74] M. Robbé and M. Sadkane. Exact and inexact breakdowns in the block GMRES method. *Linear Algebra and its Applications*, 419 :265–285, 2006.
- [75] J.D. Roberts. Linear model reduction and solution of the algebraic Riccati equation by use of the sign function. *Internat. J. Control*, 32 :677–687, 1971.
- [76] Y. Saad. Numerical solution of large Lyapunov equations. *Signal Processing, Scattering, Operator Theory and Numerical Methods*, pages 503–511, 1990.
- [77] Y. Saad. *Iterative Methods for Sparse Linear Systems (2nd ed)*. SIAM, Philadelphia, PA, 2003.
- [78] M. Sadkane. Block Arnoldi and Davidson methods for unsymmetric large eigenvalue problems. *Numer. Math.*, 64 :687–706, 1993.

-
- [79] M. Sadkane. A low-rank Krylov squared Smith method for large-scale discrete-time Lyapunov equations. *Linear Algebra and its Applications*, 8 :2807–2827, 2012.
- [80] V. Simoncini. A new iterative method for solving large-scale Lyapunov matrix equations. *SIAM J. Sci. Comput.*, 29 :1268–1288, May 2007.
- [81] J.J. Slotline and L. Weiping. *Applied Nonlinear Control*. Prentice Hall, 1991.
- [82] R. Smith. Matrix equation $XA+BX = C$. *SIAM J. Appl. Math.*, 16 :198–201, 1968.
- [83] J.C. Strikwerda. *Finite Difference Schemes and Partial Differential Equations, 2nd edition*. SIAM, 2004.
- [84] P Van Dooren. A generalized eigenvalue approach for solving Riccati equations. *SIAM J. Sci. Statist. Comput.*, 2 :121–135, 1981.
- [85] E. Wachspress. Iterative solution of the Lyapunov matrix equation. *Appl. Math. Lett.*, (1) :231–247, 1988.
- [86] H. Wendland. Piecewise polynomial, positive definite and compactly supported radial basis functions of minimal degree. *Adv. Comput. Math.*, 4 :389–396, 1995.
- [87] H. Wendland. *Scattered Data Approximation*. Cambridge University Press, 2004.
- [88] W.M. Wonham. On a matrix Riccati equation of stochastic control. *SIAM J. Contr.*, 6 :681–697, 1968.
- [89] Z. Wu. Multivariate compactly supported positive definite radial functions. *Adv. Comput. Math.*, 4 :283–292, 1995.