ROYAUME DU MAROC
Université Mohammed V - RABAT -
جامعة محمد الخامس – الرباط –
Faculté des sciences
كلية العلوم
CENTRE D'ETUDES DOCTORALES - SCIENCES ET TECHNOLOGIES

N° d'ordre : CT47

# THESE

En vue de l'obtention du : **DOCTORAT**

**Centre de recherche** : *Centre de recherche et d'applications mathématiques de Rabat*

**Structure de Recherche** : *Laboratoire de Mathématiques, informatique et applications et sécurité de l'information*

**Discipline** : *Mathématiques appliquées*

**Spécialité** : *Analyse numérique*

**Présentée et soutenue le : *15/07/2023* par :**

***Fatima Bouyghf***

## *Une approche unifiée pour l'étude des méthodes de sous-espace de Krylov pour la résolution des systèmes linéaires*

### *Devant le Jury*

| | | |
|---|---|---|
| Khalide JBILOU | PES, Université du Littoral Côte d'Opale - France | Président |
| Jilali ABOUIR | PES, Université Hassan II, Faculté des sciences - Casablanca | Rapporteur / Examinateur |
| Ahmed RATNANI | PES, Université Mohammed VI Polytechnique - Ben Guerir | Rapporteur / Examinateur |
| Mohamed EL GUIDE | PH, Université Mohammed VI Polytechnique - Ben Guerir | Rapporteur / Examinateur |
| Abdeselem Hafid BENTBIB | PES, Université Cadi Ayyad, Faculté des Sciences et Techniques -Marrakech | Examinateur |
| Abderrahim MESSAOUDI | PES, Université Mohammed V, École normale supérieure- Rabat | Invité |
| Mustapha ESGHIR | PH, Université Mohammed V, Faculté des sciences - Rabat | Directeur de Thèse |
| Hassane SADOK | PES, Université du Littoral Côte d'Opale - France | Directeur de Thèse |

**Année Universitaire : 2022-2023**

**Abstract**

To solve large linear systems, iterative methods and projection methods are commonly employed. Among these methods, the Krylov subspace methods are widely utilized. The primary principle of these methods is based on the Petrov-Galerkin condition. Krylov methods involve computing an approximation of the solution to a linear system within the Krylov subspace, with the requirement that the residual is orthogonal to another subspace known as the left subspace. The choice of the left subspace leads to different variants of Krylov methods, which vary in terms of execution time, memory usage, and computational accuracy. Therefore, our research focuses on analyzing the convergence of these methods. We have contributed to this field by proposing a unified approach and a general framework to simplify the study of these methods using left inverses of the Krylov matrix. This approach is based on the fact that all Krylov methods compute the coefficients of the minimal polynomial of the system matrix for an initial residual. By leveraging mathematical tools and the properties of orthogonal projectors, we have enhanced the computational accuracy of most of these methods while preserving the same storage and execution time. Furthermore, our approach has facilitated the development of new implementations that exhibit interesting computational performance for selected methods. We have also investigated the block case of these methods in our studies.

**Keywords:** Krylov subspace, iterative methods, projection methods, linear system, orthogonal projectors, Left inverse, convergence improvement, accuracy and stability.

**Résumé**

Pour résoudre un système linéaire de grande taille, on utilise souvent des méthodes itératives et des méthodes de projection. Parmi ces méthodes, on trouve les méthodes de sous-espace de Krylov. Le principe de ces méthodes repose sur la condition de Petrov- Galerkin. En effet, les méthodes de Krylov consistent à calculer une approximation de la solution d'un système linéaire dans le sous-espace de Krylov, à condition que le résidu soit orthogonal à un autre sous-espace, appelé sous-espace à gauche. Le choix du sous-espace à gauche donne différentes variantes des méthodes de Krylov, qui diffèrent les unes des autres en termes de temps d'exécution, de stockage en mémoire et de précision de calcul. Notre axe de recherche porte donc sur l'amélioration de la convergence de ce type de méthodes. Nous avons contribué en proposant une approche unifiée et un cadre général pour simplifier l'étude de ces méthodes en utilisant les inverses à gauche. Cette approche repose sur le fait que toutes les méthodes de Krylov calculent les coefficients du polynôme minimal de la matrice du système pour un résidu initial. En utilisant des outils mathématiques et des propriétés des projecteurs orthogonaux, nous avons pu améliorer la précision de calcul de la plupart de ces méthodes tout en conservant le même stockage et le même temps d'exécution. Grâce à notre approche, nous avons également proposé de nouvelles implémentations qui offrent des performances de calcul intéressantes pour certaines méthodes. Le cas par bloc de ces méthodes a également été étudié.

**Mots-clés:** Sous-espace de Krylov, méthodes itératives, méthodes de projection, système linéaire, projecteurs orthogonaux, inverse à Gauche, amélioration de la convergence, précision et stabilité.

**Dédicace**

Ah les dédicaces . . .! J'ai bien dû y penser une centaine de fois durant ma thèse. J'avais hâte d'en arriver à ce moment-là et d'écrire les mots justes pour chacun d'entre vous. Et bien finalement, c'est une tâche bien plus difficile à réaliser qu'on ne le croit, aucun article pour s'appuyer, seulement moi et le souvenir de ces quatre années de thèse . . . Enfin, les mots les plus simples étant les plus forts, je dédie ce travail et j'adresse toute mon affection à ma famille, et en particulier à ma maman et à mon papa qui m'ont fait comprendre que la vie n'est pas faite que de problèmes qu'on pourrait résoudre grâce à des formules mathématiques et des algorithmes. Malgré mon éloignement depuis de (trop) nombreuses années, leur intelligence, leur confiances, leurs tendresses, leur amour me portent et me guident tous les jours. Je vous aime.

**Remerciement**

Je tiens à exprimer mes plus vifs remerciements à Monsieur Abderrahim Messaoudi et Monsieur Hassane Sadok qui furent pour moi des directeurs de thèse attentif et disponible malgré leurs nombreuses charges, leurs compétences, leurs rigueurs scientifiques et leurs clairvoyances m'ont beaucoup appris. Ils ont été et resteront des moteurs de mon travail de chercheur.

J'exprime tous mes remerciements à l'ensemble des membres de mon jury: Monsieur Khalide Jbilou Professeur à l'université de Littoral Côte d'Opale en France, Monsieur Jilali Abouir Professeur à l'université Hassane II de Casablanca au Maroc, Monsieur Ahmed Ratnani Professeur à l'université Mohammed VI Polytechnique au Maroc, Monsieur Mohamed El Guide Professeur à l'université Mohammed VI Polytechnique au Maroc, Monsieur Mustapha Esghir Professeur à l'université Mohammed V au Maroc et Monsieur Abdeslem Hafid Betbib Professeur à l'université Cadi Ayyad Marrakech.

Je remercie les membres de LabMIA-SI en particulier Madame El bernoussi Souad les membres de département de mathématiques à l'école normale supérieure de Rabat pour leurs accueil.

Je remercie la directrice du laboratoire Madame Carole Rosier, la secrétaire de LMPA Madame Isabelle Brutier Buchard ainsi que Monsieur Abderrahman Bouhamidi Professeur à l'ULCO pour m'avoir accueilli dans l'unité de recherche LMPA-UCLO Calais et de m'avoir permis de travailler dans de bonnes conditions.

J'adresse toute ma gratitude à tous mes ami(e)s et à toutes les personnes qui m'ont aidés dans la réalisation de ce travail.

# Contents

# Chapter 1

# Introduction générale (version française)

Nous souhaitons résoudre les systèmes d'équations linéaires suivants

$$Ax^{(i)} = f^{(i)}, \quad i = 1, ..., s \tag{1.1}$$

avec une même matrice $A$ de taille $N \times N$, mais avec $s$ différents second membres $f^{(i)}$ pour $i = 1, \cdots, s$. Ces $s$ systèmes linéaires peuvent être résumés sous forme de blocs comme suit

$$AX = F, \tag{1.2}$$

où $A$ est une matrice dans $\mathbb{R}^{N\times N}$, $F$ et $X$ deux matrices dans $\mathbb{R}^{N\times s}$ où $X = \left[x^{(1)}, \ldots, x^{(s)}\right]$ et $F = \left[f^{(1)}, \ldots, f^{(s)}\right]$. Des problèmes tels que (1.1) et (1.2) apparaissent dans de nombreux domaines comme la physique, l'ingénierie, la chimie, la mécanique structurelle, la biologie et bien d'autres encore [36, 53, 88]. La résolution algébrique de systèmes d'équations linéaires est l'un des problèmes les plus fréquents dans le calcul scientifique.

Pour résoudre un problème du monde réel, la première phase consiste généralement à établir un modèle mathématique du problème conduisant à des équations dont la solution devrait donner les inconnues recherchés, ou du moins des approximations utiles. Le modèle peut être discret (posé dans un espace de dimension finie) ou continu

(posé dans un espace de dimension infinie). Il peut être linéaire ou non-linéaire. De nombreux problèmes continus conduisent à des systèmes d'équations différentielles ordinaires ou aux dérivées partielles [36, 53].

Si le problème est posé dans un espace de dimension infinie, la phase suivante du processus de calcul consiste à discrétiser les équations du modèle. On obtient alors des équations discrètes dans un espace de dimension finie, qui peuvent être linéaires ou non linéaires. En général, des méthodes itératives sont être utilisées pour résoudre les équations non linéaires. Dans de nombreux cas, cela conduit à résoudre des séquences de systèmes linéaires.

La troisième phase consiste à résoudre un ou plusieurs systèmes linéaires. Ces systèmes peuvent être résolus par une variété de méthodes numériques différentes. La plupart du temps, l'objectif est de résoudre le système linéaire rapidement tant que possible avec précision à l'aide d'un ordinateur. La méthode numérique choisie pour résoudre le système linéaire doit être alors programmée le plus efficacement possible et être utilisée sur un ordinateur avec des calculs efficaces.

Nos recherches portent sur l'application des méthodes de projection pour résoudre des systèmes linéaires de grandes tailles qui proviennent généralement de la discrétisation des équations aux dérivées partielles (EDP) linéaires et non linéaires en deux ou trois dimensions. Si l'ordre $N$ de la matrice $A$ est petit, alors nous pouvons résoudre (1.1) en utilisant des méthodes directes (LU, QR et Cholesky [80, 89],..). Cependant, si $N$ est grand, les méthodes directes peuvent être coûteuses en termes de mémoire et de temps. Les méthodes itératives deviennent alors intéressantes. Parmi ces méthodes, on trouve les méthodes du sous-espace de Krylov [73, 79, 89, 105]. L'analyse de la convergence de ces méthodes est un problème difficile. Nous trouvons dans la littératures quelques livres autour de ce sujet [73, 95]. L'objectif principal de ces méthodes est de chercher une approximation de la solution d'un système linéaire dans un espace appelé sous-espace de Krylov tel que le vecteur résidu vérifie une propriété d'orthogonalité avec un autre sous-espace appelé sous-espace à gauche. Le choix de ce dernier donne toutes les différentes variantes des méthodes de Krylov. Par exemple, si le sous-espace à gauche est exactement le sous-espace de Krylov, on parle des méthodes orthogonales.

De plus, si le sous-espace à gauche est différent au sous-espace de Krylov, on trouve des méthodes obliques. Nous supposerons que l'ordre $N$ de $A$ est suffisamment grand pour ne pas garder en mémoire qu'un nombre limité de vecteurs de dimension $N$ et que tous les seconds membres sont disponibles simultanément $s \ll N$. Si $s = 1$, ces méthodes sont appelées méthodes du sous-espace de Krylov standard. Ensuite, si $s > 1$, on trouve les versions global et par blocs des méthodes de Krylov. Quelques recherches ont montré que la version par blocs est plus intéressante que la version globale en termes de vitesse de convergence et de stockage [35]. Le problème principal de ces méthodes est l'existence des pannes. Mais, grâce à leurs performances, ces méthodes connaissent un grand succès dans la communauté du calcul scientifique. Elles sont maintenant largement appliquées aux équations matricielles de grande échelle (Lyapunov, Sylvester, etc [70]). Ces méthodes sont intéressantes car elles sont moins coûteuses mais elles souffraient du problème de panne lors de leur implémentation. Ce problème a été finalement résolu en utilisant une nouvelle approche de la méthode de Lanczos et des algorithmes optimaux au niveau de coût de calcul et de mémoire et sans avoir le problème de panne. On cite à titre d'exemples les références [73, 95].

Si la matrice $A$ est symétrique et définie positive, les méthodes les plus populaires pour résoudre le système linéaire sont les méthodes du gradient conjugué (CG [100]), Lanczos [59] et MINRES (MINimal RESidual [40]) et leurs variantes. Cependant, lorsque la matrice est non symétrique, de nombreuses généralisations de la méthode du gradient conjugué ont été données au cours des trente dernières années. On trouve par exemple, la méthode d'Arnoldi FOM (Full Orthogonalization method [95, 69]), GMRES (Generalized Minimal RESidual [101, 69]) et leurs variantes. Il y a aussi BiCG (Bi-Conjugate Gradient), QMR (Quasi-Minimal Residual [38]), CGS (Conjugate Gradient Squared [83]) et BiCGStab (Bi-Conjugate Gradient Stable [104]) qui sont des extensions de la méthode de Lanczos dans le cas non-symétrique. Il existe une autre famille de méthodes itératives appelées méthodes IDR (Induced Dimension Reduction) introduites dans [97]. L'approche de cette famille est différente de celle des méthodes de Krylov déjà mentionnées [91, 92] .

Si $s > 1$, nous avons la version global ou par blocs des méthodes de Krylov. Le

principe de ce type de méthode est similaire au cas standard, c'est-à-dire le cas où $s = 1$. Il s'agit de trouver une approximation de (1.2) dans un sous-espace appelé sous-espace de Krylov global ou par blocs tel que la matrice résidu vérifie une condition d'orthogonalité.

Nous avons contribué dans ce domaine, en développant une approche unifiée des méthodes de Krylov, en améliorant la convergence de certaines de ces méthodes, et en donnant une nouvelle implémentation de certaines d'entre elles. Nous avons proposé un formalisme général pour unifier et étudier les méthodes du sous-espace de Krylov d'une manière très efficace. Cela nous a permis de donner de l'importance à certains nouveaux résultats. Nous avons également étudié la version par blocs de certaines méthodes de Krylov et nous avons proposé une amélioration de leur convergence. Les sujets de recherche tournent autour des méthodes du sous-espace de Krylov (analyse de convergence et implémentation) et la résolution de systèmes d'équations linéaires avec plusieurs seconds membres. Ensuite, comme il est expliqué précédemment, cette thèse fait partie d'un effort global pour améliorer la convergence des méthodes itératives afin de:

- Développer une approche unifiée pour les méthodes de Krylov,
- Améliorer la convergence de certaines méthodes de Krylov,
- Développer des nouvelles implémentation pour des méthodes existent.

Pour atteindre cet objectif, nous étudions en détail les versions standards et par blocs de la plupart des méthodes du sous-espace de Krylov et les méthodes IDR(s). Ce manuscrit comporte une introduction générale et trois chapitres.

Dans cet article, on va présenter un cadre général pour l'étude des méthodes de sous-espace de Krylov utilisées pour résoudre le système linéaire $Ax = f$. Ces méthodes visent à atteindre la convergence dans un nombre spécifié d'itérations, noté $m$, étant donné un vecteur d'estimation initial particulier $x_0$ et son résidu correspondant $r_0 = f - Ax_0$. Notre analyse porte sur le polynôme minimal $\Phi_m$ de degré $m$ de $A$ pour le vecteur $r_0$. Nous établissons que ces méthodes englobent les méthodes de Petrov-Galerkin et les méthodes de seminormes minimales en tant que cas particuliers. De

plus, nous démontrons que les méthodes de seminormes minimales satisfont les conditions implicites de Petrov-Galerkin.

Nous fournissons une formulation générale pour les itérés basée sur des inverses généralisés. Le choix d'un inverse gauche spécifique et la méthode de construction de la base de Krylov sont des facteurs de distinction cruciaux entre les différentes méthodes de sous-espace de Krylov. Nous décrivons et analysons les propriétés mathématiques de ces méthodes, en soulignant leur dépendance à l'égard de deux matrices. Nous prouvons notamment que CMRH et QMR, en tant qu'exemples spécifiques, satisfont également aux conditions d'orthogonalité implicites de Petrov-Galerkin.

En outre, nous explorons des techniques permettant d'améliorer le comportement de convergence de ces méthodes en sélectionnant soigneusement les vecteurs dans leurs implémentations. Grâce à notre étude, nous visons à approfondir la compréhension des méthodes de sous-espace de Krylov, à donner un aperçu de leurs propriétés de convergence et à identifier des améliorations potentielles.

Nous considérons également certaines méthodes de Krylov qui sont des méthodes de produit. Dans ce cas, le $k$th résidu $r_k$ associé à l'approximation $x_k$ de la solution exacte est donné par $r_k = \Psi_k(A)\Phi_k(A)r_0$, et $\Psi_k$ est un polynôme de degré fixe ou variable. Nous examinerons des choix particuliers de $\Psi_k$ impliquant la convergence locale, le lissage, la mémoire fixe et le coût de chaque itération. Nous donnerons également une amélioration de certaines méthodes de produits telles que CGS. Pour illustrer la performance des algorithmes dérivés, nous fournissons quelques exemples numériques.

Dans le chapitre 2, nous étudions la version par bloc de quelques méthodes de sous-espace de Krylov pour résoudre le système (1.2). Nous proposons une amélioration de la convergence de ces méthodes en utilisant des projecteurs orthogonaux. Pour cela, nous rappelons quelques définitions et propriétés dans le cas par bloc, à savoir le sous-espace de Krylov par bloc. De plus, pour donner de l'importance à nos nouveaux algorithmes, nous rappelons également la méthode GMRES par bloc Bl-GMRES basée sur le processus Bl-Arnoldi car elle est la plus optimale en terme de précision. Des exemples numériques sont proposés pour comparer les nouvelles méthodes et la méthode GMRES et pour illustrer les performances de notre technique.

Enfin, le chapitre 3 est organisé comme suit, après une petite introduction, nous donnons dans la section 2, un bref aperçu de l'approche IDR et nous la comparons mathématiquement avec les méthodes du sous-espace de Krylov en expliquons la raison pour laquelle nous ne pouvons pas inclure la méthode IDR dans l'approche des méthodes de sous-espace de Krylov. Ensuite, nous proposons une amélioration de la convergence de l'algorithme IDR en utilisant des projecteurs orthogonaux. La section 3 est consacrée à la comparaison de l'algorithme proposé avec la fameuse méthode GMRES. De plus, nous présentons la version par bloc de la méthode IDR (Bl-IDR) en donnant une amélioration de la convergence. Nous développons la version global de la méthode IDR nommée pour Gl-IDR . De plus, nous proposons une amélioration qu'on va la comparer avec la méthode Gl-GMRES. Enfin, dans la dernière section, nous présentons quelques exemples numériques pour illustrer l'efficacité de l'algorithme dérivé.

# Chapter 2

# General introduction

We are interested in solving the following multiple systems of linear equations

$$Ax^{(i)} = f^{(i)}, \quad i = 1, ..., s \tag{2.1}$$

with a same matrix $A$ of size $N \times N$, and $s$ different right-hand sides $f^{(i)}$ for $i = 1, \cdots, s$. If all of the right hand sides are available simultaneously, these $s$ linear systems (2.1) can be summarized in a block form as follows

$$AX = F, \tag{2.2}$$

with $F$ and $X$ two matrices in $\mathbb{R}^{N \times s}$ where $X = \left[x^{(1)}, \ldots, x^{(s)}\right]$ and $F = \left[f^{(1)}, \ldots, f^{(s)}\right]$. Problems of the form (2.1) and (2.2) appears in many fields of applications such as physics, engineering, chemistry, structural mechanics computation, biology [36, 53, 88] and many others. Then, solving systems of linear algebraic equations is one of the most frequent problems in scientific computing.

When one wants to solve a real-world problem, usually the first phase is to set up a mathematical model of the problem leading to equations whose solution should give the quantities (also known as variables or unknowns) that are sought, or at least optimal approximations. Generally, the model can be discrete (posed in a finite-dimensional space) or continuous (posed in an infinite-dimensional space). Also, it can be linear or nonlinear. Many continuous problems lead to systems of ordinary differential or partial

differential equations [36, 53].

If the problem is posed in an infinite dimensional space, the next phase of the computational process consists of discretizing the model equations, which leads to discrete equations in a finite dimensional space, that can be linear or nonlinear. In general, an iterative method may be used to solve the nonlinear equations. In many cases, this leads to solve sequences of linear systems.

The third phase consists of solving one or more linear systems. These systems can be solved by a variety of different numerical methods. The best are those which preserve the precision of the solution and the speed of the calculations.

Our research focuses on the application of projection methods to solve large linear systems that typically arise from the discretization of linear and nonlinear partial differential equations (PDEs) in two or three dimensions. If the order $N$ of the matrix $A$ is small, then we can solve (2.1) using direct methods (LU, QR and Cholesky [80, 89],..). However, with large value of $N$, direct methods can be expensive in terms of memory and time. Iterative methods, then become appealing. Among these methods, we find the Krylov subspace methods also known as Krylov methods [63, 73, 79, 105]. The analysis of the convergence of these methods is a very difficult problem that has been developed by several researchers. The major main of these methods is to search an approximation of the solution of a linear system in a space called Krylov subspace such that the residual vector satisfy an orthogonality property lies to another subspace called left subspace. The choice of this latter gives all different variants of the Krylov methods. For example if we have the left subspace is the Krylov subspace, we talk about orthogonal type method. In the other hand, if the left subspace is different to the Krylov subspace, we find oblique methods. We will assume that the order $N$ of $A$ is sufficiently large to keep in memory only a limited number of vectors of dimension $N$ and that all right-hand sides are available simultaneously $s \ll N$. In the case where $s = 1$, these methods are called standard Krylov subspace methods. However, in some practical problems, such as in electromagnetism and signal processing [88], one has to solve several linear systems with the same matrix and several second members. Instead of solving these systems separately, it is interesting and less costly to solve them in

blocks. These methods have been developed into the block Krylov subspace methods and in particular the global methods. Next, it is shown in [35] that the block version is more interesting than the global version in terms of convergence speed. The main issue in this methods is when breakdowns exist. But, thanks to their performances, these methods are very successful in the scientific computing community. They are now widely applied in large-scale general matrix equations (Lyapunov, Sylvester, etc). These methods are interesting because of their low cost but they suffered from the breakdown problem during their implementation. This problem was finally solved using a new approach of the Lanczos method and optimal algorithms (cost and memory) without breakdown were proposed. There already exist excellent books describing Krylov methods, see for example [73, 95].

When the matrix $A$ is symmetric positive definite, the most popular methods to solve the linear system are the conjugate gradient (CG [100]), Lanczos [59] and MINRES (MINimal RESidual [40]) methods and their variants. When the matrix is non-symmetric, many generalizations of the conjugate gradient method have been given in the last thirty years. We find for example, the Arnoldi's method FOM (Full Orthogonalization method [95, 69]), GMRES (Generalized Minimal RESidual [101, 69]) and their variants. There are also BiCG (Bi-Conjugate Gradient), QMR (Quasi-Minimal Residual [38]), CGS (Conjugate Gradient Squared [83]) and BiCGStab (Bi-Conjugate Gradient Stable [104]) which are the extension of the Lanczos method in non-symmetric case. There is another family of iterative methods named IDR (Induced Dimension Reduction) methods introduced in [97] and studied in [91, 92] whose approach is different to Krylov methods.

In the case where $s > 1$, we have the block version of Krylov methods. Same as in the standard case, the main goal is finding an approximation of (2.2) in a subspace named block Krylov subspace such that the residual matrix in a left block or global subspace.

We have contributed in this area, by developing a unified approach to the Krylov methods by improving the convergence of some of these methods, and by giving a new implementation of some of them. We have proposed a general framework to unify and

study the Krylov subspace methods in a very efficient way. This allowed us to give importance to some new results. We are also studied the block version of some Krylov methods and we proposed an enhancement of their convergence and developed a new implementation to avoid breakdown problems. The research topics revolve around Krylov subspace methods (convergence analysis and implementation) and solving systems of linear equations with several right hand sides. Then, as previously explained, this thesis is part of a global effort to improve the convergence of iterative methods in order to:

- Develop a unified approach to Krylov methods,
- Improve the convergence of some Krylov subspace methods,
- Develop some new implementation to avoid breakdown problems.

To achieve this purpose, we study in detail the most variants of Krylov subspace methods in standard and block versions.The outline on the manuscript is as follow. In this chapter we have presented a comprehensive framework for studying Krylov subspace methods, explored their mathematical properties and convergence behaviour, and discussed techniques to enhance their performance. The paper covers various aspects including the minimal polynomial of matrix $A$, the relationship between different methods, the role of generalized inverses, and the use of product methods. The provided numerical examples further support the analysis and conclusions of the research.

We introduce a comprehensive framework for studying Krylov subspace methods used to solve linear systems of the form $Ax = f$, where $A$ is a matrix, $x$ is the unknown vector, and $f$ is the right-hand side vector. The objective of these methods is to achieve convergence within a specified number of iterations, denoted as $m$.

The minimal polynomial $\Phi_m$ of matrix $A$, associated with the initial residual $r_0 = f - Ax_0$, is a key focus of analysis in the paper. The degree of $\Phi_m$ is $m$, and the properties of this minimal polynomial play a crucial role in the convergence behaviour of the Krylov subspace methods.

We establish that Petrov-Galerkin methods and minimal seminorm methods are specific cases of the broader framework of Krylov subspace methods. Additionally, it is

demonstrated that minimal seminorm methods satisfy implicit Petrov-Galerkin conditions.

In this chapter, we present a general formulation for the iterates of Krylov subspace methods based on generalized inverses. The choice of a specific left inverse and the construction method of the Krylov basis are important factors that differentiate various Krylov subspace methods. The mathematical properties of these methods are described and analysed, with emphasis on their dependency on two matrices.

The chapter proves that specific instances of Krylov subspace methods, such as CMRH (Conjugate Minimum Residual with Hessenberg matrix) and QMR (Quasi-Minimal Residual), satisfy implicit Petrov-Galerkin orthogonality conditions.

Techniques for improving the convergence behavior of Krylov subspace methods by carefully selecting vectors in their implementations are explored. The aim is to deepen the understanding of these methods, provide insights into their convergence properties, and identify potential enhancements.

We also discuss Krylov methods that are product methods, where the $k$th residual $r_k$ associated with the approximation $x_k$ of the exact solution is expressed as $r_k = \Psi_k(A)\Phi_k(A)r_0$. Here, $\Psi_k$ is a polynomial of fixed or variable degree. Specific choices of $\Psi_k$, including local convergence, smoothing, fixed memory, and cost considerations for each iteration, are examined.

Enhancements of product methods such like CGS (Conjugate Gradient Squared) is presented in the paper.

In conclusion, we present a comprehensive framework for studying Krylov subspace methods, investigates their mathematical properties and convergence behaviour, explores techniques for improvement, and provides numerical examples to demonstrate the effectiveness of the proposed algorithms.

In Chapter 2, we present our technique to improve the convergence of the block version of some Krylov methods for solving non-symmetric linear systems of equations with multiple right-hand sides. This technique is similar to our technique in standard case, we apply an orthogonal projector to the residual matrix to minimize its norm. The considered method are block BiCG (Bl-BiCG [74]) and block BiCGStab

(Bl-BiCGStab [58]) methods. Then, we first give a reminder of all these methods as well as the definition of the block Krylov subspace. Secondly, To show the performance of our derived algorithms, we give a comparison with the block GMRES method (Bl-GMRES [35]) since it is the most optimal method in the level of accuracy. Finally, some numerical examples are proposed to illustrate the performance of our technique.

The remainder of the last chapter is organized as follow, after an introduction, we give in section 2, a brief review of IDR approach and compare it mathematically with Krylov subspace methods and give the reason why we can not include IDR method in Krylov subspace approach. Next, we will propose an enhancement of the convergence of IDR algorithm using orthogonal projectors. The subject of section 3 is to develop the global version of IDR method and give a enhancement of this new method. Moreover, we discuss also the possibility to improve the convergence of the block version of this method. In the last section, we will present some numerical experiments to illustrate the effectiveness of the derived algorithm compared with the standard, the global and the block GMRES method.

# Chapter 3

# A unified approach to Krylov subspace methods for solving linear systems


## Abstract

In this paper, we present a comprehensive framework for studying Krylov subspace methods used to solve the linear system $Ax = f$. These methods aim to achieve convergence within a specified number of iterations, denoted by $m$, given a particular initial estimate vector $x_0$ and its corresponding residual $r_0 = f - Ax_0$. Our analysis focuses on the minimal polynomial $\Phi_m$ of degree $m$ of $A$ for the vector $r_0$ . We establish that these methods encompass Petrov-Galerkin methods and minimal seminorm methods as special cases. Additionally, we demonstrate that minimal seminorm methods satisfy implicit Petrov-Galerkin conditions.

We provide a general formulation for the iterates based on generalized inverses. The choice of a specific left inverse and the method of constructing the Krylov basis are crucial distinguishing factors among different Krylov subspace methods. We describe and analyze the mathematical properties of these methods, emphasizing their dependency on two matrices. Notably, we prove that CMRH and QMR, as specific instances, also satisfy implicit Petrov-Galerkin orthogonality conditions.

Furthermore, we explore techniques to improve the convergence behavior of these methods by carefully selecting vectors in their implementations. Through our investigation, we aim to deepen the understanding of Krylov subspace methods, provide insights into their convergence properties, and identify potential enhancements.
We also consider some Krylov methods, which are product methods. In this case the $k$th residual $r_k$ associated with the approximation $x_k$ of the exact solution is given by $r_k = \Psi_k(A)\Phi_k(A)r_0$, and $\Psi_k$ is a polynomial of fixed or variable degree. We will examine particular choices of $\Psi_k$ involving local convergence, smoothing, fixed memory, and cost for each iteration. We will also give an enhancement of some products methods such as CGS. To illustrate the performance of the derived algorithms, we provide some numerical examples.

## 3.1 Introduction

Many problems in science and engineering require the solution of systems of linear equations. Preconditioned Krylov subspace methods appear to be particularly suited to solve linear systems when the matrix is sparse.

We consider the iterative solution of the linear system

$$Ax = f, \tag{3.1}$$

where $A$ is a real $n \times n$ nonsingular matrix and $f$ is a given vector of $\mathbb{R}^n$.
The classical Krylov subspace methods are often defined by an orthogonality or quasi-orthogonality conditions for residuals or by minimal or semi-minimal residuals conditions. The main difference between the many Krylov methods for solving linear systems is in the choice of the construction of the Krylov basis and on the choice of a left inverse, which characterizes the orthogonality or quasi-orthogonality conditions. We will prove that many properties of Krylov subspaces methods can be obtained and described in a general framework using generalized inverses.

In 1950, Lanczos proposed a method for transforming a matrix into a similar tridiagonal one. Since, by the theorem of Cayley-Hamilton, the computation of the characteristic polynomial of a matrix and the solution of a system of linear equations are equivalent problems, Lanczos [59], in 1952, used his method for that purpose.

Because of their many advantages, Krylov subspace methods have been the subject of a great deal of research, and several algorithms have been obtained for their implementation. These include Hestenes and Stiefel's famous conjugate gradient algorithm [50] when the matrix is Hermitian, and Fletcher's bi-conjugate gradient (BiCG) algorithm [37] in the general case. For non-symmetric systems, the most commonly used Krylov subspace methods are the FOM method, the GMRES method [84], and the BiCGStab method [104]. The unknown parameters can be obtained for BiCG and GMRES by imposing a Petrov-Galerkin condition (residual $r_k$ is orthogonal to a subspace of dimension $k$). Moreover, we know that the GMRES method satisfies a minimal residual condition. Other Krylov subspace methods constructed without imposing an explicit Petrov-Galerkin condition, but using instead a semi-norm residue minimization (QMR) method presented in [38], which has low storage (in general) and the CMRH (minimum changing residue method based on Hessenberg's algorithm) presented in [94]. BiCG, GMRES, FOM, QMR and CMRH can be implemented using a factorization of the Krylov matrix [73, chapter 6]. FOM and GMRES use Arnoldi's algorithm, while CMRH uses Hessenberg's algorithm [69]. Then the BiCG and QMR methods use the Lanczos algorithm. We can have a possible breakdown and we need to use an anticipated Lanczos algorithm to avoid them [38, 7, 8, 9].

Let $x_0$ be the initial approximation of the exact solution,

$x^* = A^{-1}f$, of the system (4.1) and $r_0 = f - Ax_0$ be the corresponding residual. We recall the definition of the Krylov subspace [89, 105]

**Definition 3.1.** The Krylov subspace of dimension $k$ associated to the matrix $A$ and the vector $v$ is defined by

$$K_k(A, v) = span\{v, Av, \ldots, A^{k-1}v\}.$$

Classical Krylov subspace methods [89, 105, 73] compute the approximate solution $x_k$ and its correspond residual $r_k = f - Ax_k$ such that

$$x_k - x_0 \in K_k(A, r_0), \quad \text{and } r_k = \Phi_k(A)r_0 \quad \text{for } k = 1, \ldots, m,$$

where $\Phi_k$ is a polynomial of degree $k$.

Let $K_k$ be the matrix defined by $K_k = [r_0, Ar_0, \ldots, A^{k-1}r_0]$ and $W_k = AK_k$. Then

$$x_k - x_0 = K_k d_k, \quad \text{and } r_k = r_0 - W_k d_k = \Phi_k(A)r_0 \quad \text{for } k = 1, \ldots, m.$$

In order to determine the unknown vector $d_k$, other conditions are needed, which explains why there are several Krylov subspace methods.

In the next section, we will discuss the convergence of Krylov subspace methods, which typically converge after $m$ iterations with the relation $r_k = \Phi_k(A)r_0$ for $k \leq m$. Here, $\Phi_m$ represents the minimal polynomial of $A$ for the vector$r_0$. We will begin by providing the general expression of the residual. Furthermore, we will explore how to characterize classical Krylov subspace methods using the left inverses of the matrix $W_k$ and establish the dependence of this characterization on two matrices, $Y_k$ and $Z_k$.

Additionally, we will conduct a detailed examination of various classical Krylov subspace methods. In Section 3, we will present a technique for selecting the parametric matrix $Z_k$ to enhance the convergence of these methods. We will also consider some Krylov product methods and illustrate the algorithm of selecting the polynomial $\Psi_k$ for using these methods efficiently. Moreover, we will introduce an improved version of the CGS method [83], which is one of the methods under consideration.

To demonstrate the efficacy of the derived algorithms, we will provide several examples in the concluding section.

Throughout this paper $H^\dagger$ denotes the pseudo inverse of a nonsingular square matrix $H$. The matrix $I_k$ is the identity matrix of size $k$ and the vector $e_i$ its $i$th column. we also use the notation $H_{k+1,k}$ for a rectangular matrix with $k+1$ rows and $k$ columns.

For simplicity of the exposition, throughout the paper we assume exact arithmetic and

real data.

## 3.2 Preliminary results

In this section we will use the general left inverse of $W_k$ [13], for characterizing the Krylov subspace methods. This characterization depends on two parameters $Y_k$ and $Z_k$, which are two $n \times k$ matrices to be chosen for leading to Krylov subspace methods. Then, we show how to improve the convergence of the BiCG method [63, 73, 89, 105].

### 3.2.1 Characterization of Krylov subspace methods

Let

$$\Phi_m(\xi) = \sigma_0 + \sigma_1\xi + \cdots + \sigma_m\xi^m = \sum_{i=0}^{m} c_i\xi^i, \quad \text{with} \quad \Phi_m(0) = 1,$$

be the minimal polynomial of the matrix $A$ for the vector $r_0$, i.e.

$$\Phi_m(A)r_0 = \sum_{i=0}^{m} \sigma_i A^i\, r_0 = 0, \tag{3.2}$$

and

$$m = \min\{k \quad \text{such that} \quad \sum_{i=0}^{k} \sigma_i A^i r_0 = 0, \quad \text{with} \quad \sigma_0 = 1\}.$$

Let $w_i$ be the vectors defined by

$$w_0 = r_0, \quad w_i = A^i r_0, \quad \text{for} \quad i = 1, \cdots, m,$$

and $\widetilde{c}$ be the vector whose components are $-\sigma_1, \ldots, -\sigma_m$. If we set $W_m = [w_1, \ldots, w_m]$, then the relation (4.2) can be written in matrix form as

$$W_m\widetilde{c} = r_0. \tag{3.3}$$

It is important to remark that the system (3.3) has a unique solution and that the rank of $W_m$ is $m$.

We first recall the notion of general left inverses of $W_m$. Let $W_m^L$ be a general left inverse of $W_m$ [13], i.e., $W_m^L W_m = I_m$, then it has been shown that if $W_m^\ell$ is a particular left inverse of $W_m$, a general left inverse $W_m^L$ can be given by [13]

$$W_m^L = W_m^\ell + Z_m^T \left( I_n - W_m W_m^\ell \right), \tag{3.4}$$

where $Z_m$ is an arbitrary $n \times m$ matrix.

As the rank of $W_m$ is $m$ ($W_m$ is a full rank matrix), there exists an arbitrary $n \times m$ matrix $Y_m$ of rank $m$ such that $Y_m^T W_m$ is nonsingular (for example $Y_m = W_m$). So, we define a particular left inverse $W_m^\ell$ of the matrix $W_m$ by

$$W_m^\ell = (Y_m^T W_m)^{-1} Y_m^T. \tag{3.5}$$

Therefore, by using the general and particular left inverses, the linear system (3.3) can be solved in two distinct ways. Thus, (3.3) becomes

$$r_0 - W_m W_m^L r_0 = 0.$$

From the general left inverse $W_k^L$, with $k \leq m$, the general residual vector $r_k^K$ and the approximate solution $x_k^K$ are defined by

$$r_k^K = (I - W_k W_k^L) r_0 = f - A x_k^K, \tag{3.6}$$

with

$$r_0^K = r_0, \quad x_m^K = A^{-1} b = x^*, \quad r_m^K = 0.$$

Therefore, we obtain for $k = 1, \cdots, m-1$

$$x_k^K = x_0 + K_k W_k^L r_0, \tag{3.7}$$

where $K_k = A^{-1}W_k$ is the Krylov matrix whose columns are $w_0, \cdots, w_{k-1}$. Using the relations (3.4), (3.5) and (3.6) we get

$$\begin{aligned} r_k^K &= (I_n - W_k W_k^L) r_0 \\ &= \left(I_n - W_k[W_k^\ell + Z_k^T(I_n - W_k W_k^\ell)]\right) r_0 \\ &= (I_n - W_k Z_k^T)(I_n - W_k W_k^\ell) r_0. \end{aligned} \tag{3.8}$$

Let $Y_k$ be the matrix whose columns are denoted by $y_i$, for $i = 1, \cdots, k$. We define

$$\mathcal{Y}_k = span\{y_1, \cdots, y_k\}.$$

By setting $Z_k = 0_{n\times k}$ or $Z_k = (W_k^\ell)^T$, and choosing the matrix $Y_k$, we can obtain most of the Krylov subspace methods. The residuals $r_k^P$ obtained by using the left inverse $W_k^\ell$ are mathematically equivalent to the residuals of the known Petrov-Galerkin methods defined by the Petrov-Galerkin condition, which consists in imposing that the residual $r_k^P$ is orthogonal to $\mathcal{Y}_k$, that is,

$$x_k^P - x_0 \in K_k(A, r_0), \text{ and } r_k^P = f - Ax_k^P \perp \mathcal{Y}_k.$$

Hence $r_k^P = \left(I_n - W_k(Y_k^T W_k)^{-1} Y_k^T\right) r_0$, and $Y_k^T r_k^P = 0$.

**Theorem 3.2.** *The kth residual in Krylov subspace methods defined by*

$$r_k^K = f - Ax_k^K, \ x_k^K - x_0 \in K_k(A, r_0), \text{ and } r_m^K = 0,$$

*where m is the degree of the minimal polynomial of the matrix A for the vector $r_0$, can be written, $\forall k \in \{1, 2, \ldots, m-1\}$, as*

$$r_k^K = \left(I_n - W_k Z_k^T\right)\left(I_n - W_k(Y_k^T W_k)^{-1} Y_k^T\right) r_0 = \left(I_n - W_k Z_k^T\right) r_k^P,$$

*where $Y_k$ is an arbitrary $n \times k$ matrix such that $Y_k^T W_k$ is invertible and $Z_k$ is an arbitrary $n \times k$ matrix.*

The classical Krylov subspace methods are given in the following table, when $Z_k = 0$. For more details about the choice of the matrix $Y_k$ for the different methods see the references associated to each method.

| Method | Condition | Choice of $Y_k$ |
|---|---|---|
| FOM [89] | $K_k^T A K_k$ nonsingular | $K_k$ |
| GMRES [84, 89, 105] | $A$ nonsingular | $AK_k$ |
| Hessenberg [31] | $A$ nonsingular | $\left[e_{p_1}, e_{p_2}, \ldots, e_{p_k}\right]$ |
| Lanczos [63, 89] | $Y_k^T A K_k$ nonsingular | $\left[y, A^T y, \ldots, \left(A^T\right)^{k-1} y\right]$ |

Table 1: The choices of the matrix $Y_k$

### 3.2.2 Full Orthogonalization Method (FOM) and Generalized Minimum Residual Method (GMRES)

In this section we summarize the GMRES and Arnoldi methods. Let us first remark that

$$\left(I_n - W_k W_k^{\dagger}\right)\left(I_n - W_k W_k^{L}\right) = \left(I_n - W_k W_k^{\dagger}\right),$$

and that

$$\left(I_n - W_k W_k^{L}\right)\left(I_n - W_k W_k^{\dagger}\right) = \left(I_n - W_k W_k^{L}\right).$$

Hence if we set $Z_k$ such that $Z_k^T = W_k^{\dagger}$ and choose $Y_k$ as an arbitrary $n \times k$ matrix such that $Y_k^T W_k$ is invertible, we obtain the GMRES method

$$r_k^G = \left(I_n - W_k W_k^{\dagger}\right)\left(I_n - W_k W_k^{\ell}\right) r_0 = \left(I_n - W_k W_k^{\dagger}\right) r_0,$$

and we have

$$||r_k^G|| = \min_{z \in K_k(A, r_0)} ||b - A\,(x_0 + z)||.$$

Now, if we choose $Y_k = W_k$ and $Z_k^T = (W_k^T A^{-T} W_k) W_k^T A^{-T}$, we obtain

$$\begin{aligned} r_k^F &= \left(I_n - W_k (W_k^T A^{-T} W_k) W_k^T A^{-T}\right) \left(I_n - W_k W_k^{\dagger}\right) r_0 \\ &= \left(I_n - W_k (W_k^T A^{-T} W_k) W_k^T A^{-T}\right) r_k^G \\ &= \left(I_n - W_k (W_k^T A^{-T} W_k) W_k^T A^{-T}\right) r_0. \end{aligned}$$

The GMRES and FOM implementations are based upon the Arnoldi recursion, which corresponds to an implicit QR factorization based on the Gram-Schmidt algorithm. For the usual description of FOM and of GMRES, see [84, 89].

Consider the QR factorization of

$$K_k = V_k \tilde{R}_k,$$

where $V_k \in \mathbb{R}^{n \times k}$ is such that $V_k^T V_k = I_k$. We know that the columns of $V_k$ form an orthonormal basis of $K_k(A, r_0)$. and that $\tilde{R}_k$ is upper triangular.

Since $K_{k+1} = [r_0, AK_k] = [r_0, W_k]$, we can then write

$$K_{k+1} \begin{bmatrix} 0 \\ \\ I_k \end{bmatrix} = V_{k+1} \tilde{R}_{k+1} \begin{bmatrix} 0 \\ \\ I_k \end{bmatrix} = A\,K_k = A\,V_k\,\tilde{R}_k.$$

Since $\tilde{R}_k^{-1}$ is also upper triangular, we can define the following $(k+1) \times k$ upper Hessenberg matrix by the following relations

$$H_{k+1,k} = \tilde{R}_{k+1} \begin{bmatrix} 0_{1\times k} \\ \\ I_k \end{bmatrix} \tilde{R}_k^{-1} = V_{k+1}^T A V_k = \begin{bmatrix} H_k \\ \\ v_{k+1}^T A V_k \end{bmatrix} = \begin{bmatrix} H_k \\ \\ h_{k+1,k} \left(e_k^{(k)}\right)^T \end{bmatrix},$$

and $H_k \equiv V_k^T A V_k$ is a square upper Hessenberg matrix of dimension $k$. Moreover we have the following *Arnoldi relation*

$$A\,V_k = V_{k+1} H_{k+1,k} = V_k\,H_k + h_{k+1,k}\,v_{k+1}\,e_k^T. \tag{3.9}$$

In GMRES and FOM, the starting vector $v_1 = r_0/\|r_0\|$ and the corresponding residual vectors $r_k^F$ and $r_k^G$ are

$$\begin{aligned} r_k^F &= \left(I_n - W_k(K_k^T W_k)^{-1} K_k^T\right) r_0 \\ &= r_0 - AV_k\tilde{R}_k \left(\tilde{R}_k^T V_k^T A V_k \tilde{R}_k\right)^{-1} \tilde{R}_k^T V_k^T r_k^0 \\ &= r_0 - AV_k(V_k^T A V_k)^{-1} V_k^T r_0. \\ &= r_0 - V_{k+1} H_{k+1,k}\,(H_k)^{-1}\,V_k^T r_0, \end{aligned}$$

and

$$\begin{aligned} r_k^G &= \left(I_n - W_k(W_k^T W_k)^{-1} W_k^T\right) r_0 \\ &= r_0 - AV_k\tilde{R}_k \left(\tilde{R}_k^T (AV_k)^T A V_k \tilde{R}_k\right)^{-1} \tilde{R}_k^T (AV_k)^T r_k^0 \\ &= r_0 - AV_k((AV_k)^T A V_k)^{-1} (AV_k)^T r_0. \\ &= r_0 - V_{k+1} H_{k+1,k}\,(H_{k+1,k})^{\dagger}\,V_{k+1}^T r_0. \end{aligned}$$

In the following lemma, we give the expression of $H_k^{-1} = H_k^{\dagger}$ and $H_{k+1,k}^{\dagger}$.

**Lemma 3.3.** *Let $H_k$ be an invertible square upper Hessenberg matrix of dimension $k$ and let $H_{k+1}$ be a $(k+1) \times k$ upper Hessenberg matrix, then*

$$H_k^{-1} = [H_{k,k-1}, h_k]^{+} = \begin{bmatrix} H_{k,k-1}^{+} - H_{k,k-1}^{+} h_k q_k^T \\ \\ q_k^T \end{bmatrix},$$

*and*

$$H_{k+1,k}^{\dagger} = \begin{bmatrix} H_k \\ \\ h_{k+1,k}\left(e_k^{(k)}\right)^T \end{bmatrix}^{\dagger} = \begin{bmatrix} H_k^{-1} - \frac{h_{k+1,k}^2}{1+h_{k+1,k}^2(q_k,q_k)} H_k^{-1} q_k q_k^T & \frac{h_{k+1,k}}{1+h_{k+1,k}^2(q_k,q_k)} H_k^{-1} q_k \end{bmatrix},$$

*where* $q_k = \dfrac{(I - H_{k,k-1}H_{k,k-1}^{+})\, h_k}{||(I - H_{k,k-1}H_{k,k-1}^{+})\, h_k||^2}$.

*Proof.* . The proof of first formula is a consequence of formulas (8) and (16) in [41]. If we apply Corollary 1, page 267 of [13] to the matrix $H_{k,k+1}$, we obtain the second formula. □

A consequence of this lemma is the following theorem

**Theorem 3.4.** *Let us assume that the Arnoldi matrix $H_k = V_k^T A V_k$ is invertible, then*

1. $r_k^F = -\|r_0\|\, h_{k+1,k}(q_k, e_1) v_{k+1}$,
2. $r_k^G = \|r_0\| \left( \frac{h_{k+1,k}^2(q_k,e_1)}{1+h_{k+1,k}^2(q_k,q_k)} V_k q_k - \frac{h_{k+1,k}(q_k,e_1)}{1+h_{k+1,k}^2(q_k,q_k)} v_{k+1} \right)$,
3. $r_k^G = \left(I - AV_k(AV_k)^{\dagger}\right) r_k^F$.

*Proof.* From Lemma 3.3, we deduce the following formulas

$$H_{k+1,k}H_k^{-1} = \begin{bmatrix} I_k \\ \\ h_{k+1,k}q_k^T \end{bmatrix},$$

and

$$H_{k+1,k}H_{k+1,k}^{\dagger} = \begin{bmatrix} I_k - \frac{h_{k+1,k}^2}{1+h_{k+1,k}^2(q_k,q_k)} q_k q_k^T & \frac{h_{k+1,k}}{1+h_{k+1,k}^2(q_k,q_k)} q_k \\ \\ \frac{h_{k+1,k}}{1+h_{k+1,k}^2(q_k,q_k)} q_k^T & 1 - \frac{1}{1+h_{k+1,k}^2(q_k,q_k)} \end{bmatrix}.$$

Multipliply $H_{k+1,k}H_k^{-1}$ by $e_1^{(k)}$ and $H_{k+1,k}H_{k+1,k}^{\dagger}$ by $e_1^{(k+1)}$, we obtain

$$H_{k+1,k}H_k^{-1}e_1^{(k)} = \begin{bmatrix} e_1^{(k)} \\ h_{k+1,k}q_k^T e_1^{(k)} \end{bmatrix}, \quad \text{and} \quad H_{k+1,k}H_{k+1,k}^{\dagger}e_1^{(k+1)} = \begin{bmatrix} e_1^{(k)} - \frac{h_{k+1,k}^2 q_k^T e_1^{(k)}}{1+h_{k+1,k}^2(q_k,q_k)}q_k \\ \frac{h_{k+1,k}q_k^T e_1^{(k)}}{1+h_{k+1,k}^2(q_k,q_k)} \end{bmatrix}.$$

We can now premultiply the above vectors by $\|r_0\|V_{k+1}$ to deduce the results of the first two statements.

By subtracting $r_k^F$ from $r_k^G$, we get

$$r_k^G - r_k^F = \|r_0\| \, \frac{h_{k+1,k}^2(q_k, e_1)}{1 + h_{k+1,k}^2(q_k, q_k)} \, (V_k q_k + h_{k+1,k}(q_k, q_k)v_{k+1}) \, .$$

On the other hand

$$\begin{aligned} V_{k+1}H_{k+1,k}H_{k+1,k}^{\dagger}V_{k+1}^T r_k^F &= -\|r_0\| \; h_{k+1,k}(q_k, e_1)V_{k+1}H_{k+1,k}H_{k+1,k}^{\dagger}e_{(k+1)}^{(k+1)} \\ &= -\|r_0\| \; \tfrac{h_{k+1,k}(q_k,e_1)}{1+h_{k+1,k}^2(q_k,q_k)}V_{k+1}\begin{bmatrix} h_{k+1,k}q_k \\ h_{k+1,k}^2(q_k,q_k) \end{bmatrix}. \end{aligned}$$

We deduce that $r_k^G = (I - V_{k+1}H_{k+1,k}H_{k+1,k}^{\dagger}V_{k+1}^T)r_k^F = \left(I - AV_k(AV_k)^{\dagger}\right) r_k^F$, which ends the proof. □

### 3.2.3 Generalized Hessenberg algorithm

We describe now the Generalized Hessenberg method due to Hessenberg [107], and to Householder and Bauer [51]. This algorithm is used for the reduction of a general matrix to its Hessenbeg form. It contains the methods of Arnoldi, Lanczos and Hessenberg as particular cases.

The Generalized Hessenberg algorithm constructs a basis $\{b_1, \ldots, b_k\}$ of $K_k(A, r_0)$ by imposing an orthogonality condition on $b_{k+1}$

$$b_{k+1} \perp span\{y_1, \ldots, y_k\}.$$

Let $Y_k$ be the $n \times k$ matrix whose columns $y_1, y_2, \ldots, y_k$ are linearly independent vectors of $\mathbb{R}^n$. These two properties are verified if we first choose a non zero scalar $\gamma_1$ such that $r_0 = \gamma_1 b_1$ and define $\gamma_{k+1}$ and $b_{k+1}$ such that

$$\gamma_{k+1} b_{k+1} = \left( I - K_k (Y_k^T K_k)^{-1} Y_k^T \right) A^k r_0. \tag{3.10}$$

The non zeros scalars $\gamma_1, \ldots, \gamma_{k+1}$ are scaling vectors and can be selected in several manners depending of the choice of matrix $Y_k$. Let $B_k$ be the $n \times k$ matrix whose columns are $b_1, b_2, \ldots, b_k$, we deduce that

$$\gamma_{k+1} b_{k+1} = \left( I - B_k B_k^{\ell} \right) A b_k = \left( I - B_k (Y_k^T B_k)^{-1} Y_k^T \right) A b_k.$$

If we assume that $y_i^T b_i \neq 0$, for $i = 1, \ldots, k$, then the matrix $\tilde{L}_k \equiv Y_k^T B_k$, is an invertible lower triangular matrix and we can prove by induction [86], that the matrix $H_k^h \equiv B_k^{\ell} A B_k$ is a square Hessenberg matrix and the generalized Hessenberg relation follows

$$A B_k \;\; = \;\; B_{k+1} H_{k+1,k}^h, \tag{3.11}$$

with

$$H_{k+1,k}^h = \begin{bmatrix} B_k^{\ell} A B_k \\ \\ \delta_{k+1} \left( e_k^{(k)} \right)^T \end{bmatrix} = \begin{bmatrix} H_k^h \\ \\ h_{k+1,k}^h \left( e_k^{(k)} \right)^T \end{bmatrix}.$$

An important choice for the matrix $Y_k$ in the Generalized Hessenberg algorithm is the Krylov matrix $K_k$. This leads to the Arnoldi algorithm, since $B_k = V_k$.

In his habilitation thesis [86], the author showed how to use the Generalized Hessenberg algorithm to derive some Krylov subspace methods based on the Petrov-Galerkin condition. Since $r_0 = \gamma_1 b_1 = \gamma_1 B_k e_1^{(k)}$ and $Y_k^T B_{k+1} H_{k+1,k}^h = \tilde{L}_k H_k^h$, it holds

$$\begin{aligned} r_k^P &= \left(I_n - AK_k(Y_k^T AK_k)^{-1}Y_k^T\right) r_0 \\ &= \left(I_n - AB_k(Y_k^T AB_k)^{-1}Y_k^T\right) r_0 \\ &= \left(I_n - B_{k+1}H_{k+1,k}^h(Y_k^T B_{k+1}H_{k+1,k}^h)^{-1}Y_k^T\right) r_0 \\ &= r_0 - \gamma_1\ B_{k+1}H_{k+1,k}^h \left(H_k^h\right)^{-1} e_1^{(k)}. \end{aligned}$$

The QMR and CMRH methods, based on Generalized Hessenberg algorithm, was proposed to improve the convergence behaviour of Petrov-Galerkin methods. In [52], Heyouni and Sadok proposed the Minimizing Residual Seminorm method(MRS) and define its residual $r_k^M$ by

$$r_k^M = r_0 - \delta_1\ B_{k+1}H_{k+1,k}^h \left(H_{k+1,k}^h\right)^{\dagger} e_1^{(k+1)}, \tag{3.12}$$

and gave a relationship between $r_k^P$ and $r_k^M$ (see Theorem 2 of [52]).
In each iteration of the method, the iterate $x_k^M$ is chosen suth that its residual has minimal seminorm, i.e.,

$$|r_k^M|_{T_{k+1}} = \min_{x\in x_0+K_k(r_0,A)} |f - Ax|_{T_{k+1}} = \min_{x\in x_0+K_k(r_0,A)} \|B_{k+1}^\ell(f - Ax)\|,$$

where $|u|_{T_{k+1}} = \sqrt{u^T T_k u}$, and $T_{k+1} = \left(B_{k+1}^\ell\right)^T B_{k+1}^\ell$ is a symmetric matrix such that $u^T T_{k+1} u$ is positive for all vector $u$ in $K_{k+1}(A, r_0)$.
Let us now show that the vectors $r_k^M$ verify also an orthogonality property. We have

$$\begin{aligned} r_k^M &= r_0 - AB_k \left(B_{k+1}^\ell AB_k\right)^{\dagger} B_{k+1}^\ell r_0 \\ &= \delta_1 B_{k+1} \left(I - H_{k+1,k}^h \left(H_{k+1,k}^h\right)^{\dagger}\right) e_1^{(k+1)}. \end{aligned}$$

We can now premultiply the above equality, first by $B_{k+1}^\ell$ and after by $\left(H_{k+1,k}^h\right)^T$, to

obtain

$$\begin{aligned}\left(H_{k+1,k}^h\right)^T B_{k+1}^\ell r_k^M &= \delta_1 \left( \left(H_{k+1,k}^h\right)^T - \left(H_{k+1,k}^h\right)^T H_{k+1,k}^h \left(H_{k+1,k}^h\right)^\dagger \right) e_1^{(k+1)} \\ &= \delta_1 \left( \left(H_{k+1,k}^h\right)^T - \left(H_{k+1,k}^h\right)^T H_{k+1,k}^h \left( \left(H_{k+1,k}^h\right)^T H_{k+1,k}^h \right)^{-1} \left(H_{k+1,k}^h\right)^T \right) e_1^{(k+1)} \\ &= 0.\end{aligned}$$

This gives

$$\left((B_{k+1}^\ell)^T H_{k+1,k}^h\right)^T r_k^M = 0.$$

In addition, we remark that $\left(H_{k+1,k}^h\right)^\dagger B_{k+1}^\ell$ is also a left inverse of $AB_k$, since

$$\begin{aligned}\left(H_{k+1,k}^h\right)^\dagger B_{k+1}^\ell AB_k &= \left(H_{k+1,k}^h\right)^\dagger B_{k+1}^\ell B_{k+1} H_{k+1,k}^h \\ &= I_k.\end{aligned} \tag{3.13}$$

If we write $H_k^h = \left[H_{k,k-1}^h, h_k^h\right]$, and define $q_k^h$ by

$$q_k^h = \frac{(I - H_{k,k-1}^h (H_{k,k-1}^h)^\dagger)\, h_k^h}{||(I - H_{k,k-1}^h (H_{k,k-1}^h)^\dagger)\, h_k^h||^2},$$

we can give, in the following two theorems, the relationships between the residuals for the Petrov-Galerkin method and the MRS methods.

**Theorem 3.5.** *Let us set $r_0 = \delta_1 b_1$ and assume that in the Hessenberg algorithm the matrix $Y_k^T B_k$ is invertible. If we set $d_k = b_k - B_{k-1}(H_{k,k-1}^h)^\dagger h_k^h$, the iterates $x_k^P$, $x_k^M$ and their residual vectors $r_k^P$, $r_k^M$ are such that*

1. $x_k^{PG} - x_{k-1}^M = \delta_1\ (q_k^h, e_1)\ d_k \qquad$ *and* $\qquad r_k^P - r_{k-1}^M = -\delta_1\ (q_k^h, e_1)\ Ad_k$
2. $x_k^M - x_{k-1}^M = \frac{\delta_1 (q_k^h, e_1)}{1+(h_{k+1,k}^h)^2 (q_k^h, q_k^h)}\ d_k \qquad$ *and* $\qquad r_k^M - r_{k-1}^M = -\frac{\delta_1 (q_k^h, e_1)}{1+(h_{k+1,k}^h)^2 (q_k^h, q_k^h)}\ Ad_k,$
3. $r_k^P - r_{k-1}^M = (1 + (h_{k+1,k}^h)^2 (q_k^h, q_k^h))(r_k^M - r_{k-1}^M).$

This Theorem was proved in [52]. We remark that if the matrix $H_k$ is regular, we can write the vector $d_k$ as

$$d_k = \frac{1}{(q_k^h, q_k^h)} B_k \left(H_k^h\right)^{-1} q_k^h. \tag{3.14}$$

We have also the following results

**Theorem 3.6.** *Let us set $r_0 = \delta_1 b_1$ and assume that in the Hessenberg algorithm the matrix $Y_k^T B_k$ is invertible, then*

1. $r_k^M = \delta_1 \left( \frac{(h_{k+1,k}^h)^2 (q_k^h, e_1)}{1+(h_{k+1,k}^h)^2 (q_k^h, q_k^h)} B_k q_k^h - \frac{h_{k+1,k}(q_k^h, e_1)}{1+(h_{k+1,k}^h)^2 (q_k^h, q_k^h)} b_{k+1} \right)$.
2. *Moreover if the matrix $Y_k^T A B_k$ invertible, then*

$$r_k^P = -\delta_1 \, h_{k+1,k}^h (q_k^h, e_1) b_{k+1},$$

*and*

$$\begin{aligned} r_k^M &= r_k^P + \delta_1 \, \frac{(h_{k+1,k}^h)^2 (q_k^h, e_1)}{1+(h_{k+1,k}^h)^2 (q_k^h, q_k^h)} \left( B_k q_k^h + h_{k+1,k}^h (q_k^h, q_k^h) b_{k+1} \right) \\ &= (I - B_{k+1} H_{k+1,k}^h \left(H_{k+1,k}^h\right)^{\dagger} B_{k+1}^{\ell}) r_k^P. \end{aligned}$$

The proof of this theorem is based on lemma 3.3 and is similar to that of Theorem 3.4. From the last statement, we deduce that the methods can also be defined by the general formula given in Theorem3.2, since we can write

$$\begin{aligned} r_k^M &= (I - A B_k \left(H_{k+1,k}^h\right)^{\dagger} B_{k+1}^{\ell}) r_k^P \\ &= (I - W_k Z_k) r_k^P \\ &= (I - W_k Z_k) \left(I - W_k (Y_k^T W_k)^{-1} Y_k^T\right) r_0. \end{aligned}$$

In the next section we will examine particular cases of the Generalized Hessenberg algorithm, and we will review the Krylov subspace methods under consideration, and discuss their properties.

#### 3.2.3.1 The FOM/GMRES pair

Let us set $Y_k = K_k$. With this choice the Generalized Hessenberg algorithm reduces to the Arnoldi's algorithm and we have $B_k = V_k$ and $B_k^{\ell} = V_k^T$. Moreover the Petrov-Galerkin method becomes FOM. Finally the results of Theorem 3.4 gives a relationship

between the residuals of GMRES and FOM methods, since the matrix $V_k$ is orthogonal. These results are summarized in the following theorem

**Theorem 3.7.** *Let us assume that the Arnoldi matrix $H_k = V_k^T A V_k$ is invertible, then*

1. $\|r_k^F\| = \|r_0\| \, h_{k+1,k} |(q_k, e_1)|$,
2. $\|r_k^G\| = \|r_k^F\| \sqrt{\frac{1}{1+h_{k+1,k}^2 (q_k, q_k)}}$,
3. $\|r_k^G - r_k^F\| = \|r_k^F\| \, \sqrt{1 - \frac{1}{1+h_{k+1,k}^2 (q_k, q_k)}} = \sqrt{\|r_k^F\|^2 - \|r_k^G\|^2}$.

We remark that the residuals in FOM are orthogonal and we have $K_k^T r_k^F = 0$. The residuals in GMRES method can also defined by the orthogonality property $W_k^T r_k^G = 0$.

### 3.2.4 The Hessenberg/CMRH pair

Instead of using an implicit QR factorization as in Arnoldi algorithm, we consider here the LU factorization of the $n \times k$ Krylov matrix

$$K_k = L_k U_k, \tag{3.15}$$

with $L_k \in \mathbb{R}^{n \times k}$ a lower trapezoidal $n \times k$ matrix, and $U_k$ upper triangular. As we did in preceding subsection for the Arnoldi algorithm, we can now write

$$K_{k+1} \begin{bmatrix} 0 \\ \\ I_k \end{bmatrix} = L_{k+1} U_{k+1} \begin{bmatrix} 0 \\ \\ I_k \end{bmatrix} = A \, K_k = A \, L_k \, U_k,$$

and similarly since $U_k^{-1}$ is upper triangular, we define the following $(k+1) \times k$ upper Hessenberg matrix

$$H_{k+1,k}^h = U_{k+1} \begin{bmatrix} 0_{1 \times k} \\ \\ I_k \end{bmatrix} U_k^{-1} = \begin{bmatrix} H_k^h \\ \\ h_{k+1,k}^h \left( e_k^{(k)} \right)^T \end{bmatrix},$$

with $H_k^h$ being a square upper Hessenberg matrix of dimension $k$. We also have the following Hessenberg relation

$$\begin{aligned} A\,L_k &= L_{k+1}U_{k+1}\begin{bmatrix} 0 \\ \\ I_k \end{bmatrix} U_k^{-1} \\ &= L_{k+1}H_{k+1,k}^h = L_k\,H_k^h + h_{k+1,k}^h\,\ell_{k+1}\,e_k^T. \end{aligned} \tag{3.16}$$

Thus, the columns of $L_k$ form a different (non-orthogonal) basis of $K_k(A, r_0)$.

The Hessenberg algorithm consists of building iteratively the basis $\{\ell_1, \ldots, \ell_k\}$ of $K_k(A, r_0)$, so that $L_k = [\ell_1, \ldots, \ell_k]$ is a lower trapezoidal $n \times k$ matrix. It begins appropriately by computing $\alpha\ell_1 = r_0$, with $\alpha = (r_0)_1$ the first component of $r_0$.

By choosing $Y_k = I_k^{(n)}$ in the Generalized Hessenberg algorithm, we obtain the Hessenberg algorithm, without pivoting. Consequently the vector $\ell_{k+1}$ is defined by

$$h_{k+1,k}^h\ell_{k+1} = \left(I - L_kL_k^\ell\right)A\ell_k = \left(I - L_k((I_k^{(n)})^TL_k)^{-1}(I_k^{(n)})^T\right)A\ell_k,$$

and

$$L_k^\ell = \begin{bmatrix} (L_k^1) \\ \\ (L_k^2) \end{bmatrix}^\ell = \begin{bmatrix} ((L_k^1))^{-1} & O \end{bmatrix}.$$

This equation can be rewritten as

$$h_{k+1,k}^h\ell_{k+1} = \begin{bmatrix} 0 \\ \\ A\ell_k - (L_k^2)(L_k^1)^{-1}A\ell_k \end{bmatrix},$$

and $h_{k+1,k}^h$ is chosen such that $(\ell_{k+1})_{k+1} = 1$. Moreover

$$H_{k+1,k}^h = \begin{bmatrix} L_k^\ell A L_k \\ \\ h_{k+1,k}^h \left(e_k^{(k)}\right)^T \end{bmatrix} = \begin{bmatrix} (L_k^1)^{-1}(I_k^{(n)})^T A L_k \\ \\ h_{k+1,k}^h \left(e_k^{(k)}\right)^T \end{bmatrix} = \begin{bmatrix} H_k^h \\ \\ h_{k+1,k}^h \left(e_k^{(k)}\right)^T \end{bmatrix}.$$

To avoid breakdown and to ensure a more stable algorithm, we can use a pivoting strategy as in Gaussian elimination. We first compute the index $i_1$ such that $|(r_0)_{i_1}| = \max_{i=1,\ldots,n} |(r_0)_i|$ and set $\ell_1 = r_0/(r_0)_{i_1}$ and $y_1 = e_{i_1}^{(n)}$. Let us assume that the indexes $i_1, \ldots, i_k$ have already been obtained and set $Y_k = [e_{i_1}^{(n)}, \ldots, e_{i_k}^{(n)}]$. To obtain $i_{k+1}$ we have to compute

$$h_{k+1,k}^h \ell_{k+1} = d = \left(I - L_k((Y_k^{(n)})^T L_k)^{-1}(Y_k^{(n)})^T\right) A\ell_k,$$

and define $l_{k+1} = d/(d_{i_{k+1}})$ and $h_{k+1,k}^h = (d)_{i_{k+1}}$, where $|(d)_{i_{k+1}}| = \max_{i=1,\ldots,n} |(d)_i|$. Hence $\|\ell_{k+1}\|_\infty = 1$

By setting $Y_k = [e_{i_1}^{(n)}, \ldots, e_{i_k}^{(n)}]$, the residual for the Hessenberg method for solving the linear system is defined by

$$\begin{aligned} r_k^{Hess} &= \left(I_n - AL_k(Y_k^T AL_k)^{-1}Y_k^T\right) r_0 \\ &= r_0 - (r_0)_{i_1}\, L_{k+1} H_{k+1,k}^h \left(H_k^h\right)^{-1} e_1^{(k)}. \end{aligned}$$

The CMRH method is a Minimizing Residual Seminorm methods and its $k$th residual is defined by

$$\begin{aligned} r_k^C &= r_0 - (r_0)_{i_1}\, L_{k+1} H_{k+1,k}^h \left(H_{k+1,k}^h\right)^\dagger e_1^{(k+1)} \\ &= \left(I_n - AL_k\left(\left(H_{k+1,k}^h\right)^T L_{k+1}^\ell A L_k\right)^{-1}\left(H_{k+1,k}^h\right)^T L_{k+1}^\ell\right) r_0. \end{aligned}$$

We have the two orthogonality conditions

$$[e_{i_1}^{(n)}, \ldots, e_{i_k}^{(n)}]^T r_k^h = 0, \qquad \text{and} \qquad \left(H_{k+1,k}^h\right)^T L_{k+1}^\ell r_k^C = 0.$$

Moreover $\left(H^h_{k+1,k}\right)^\dagger L^\ell_{k+1}$ is a left inverse of $AL_k = H^h_{k+1,k}L_{k+1}$, and we have

$$\begin{aligned} r^C_k &= \left(I_n - AL_k\left(H^h_{k+1,k}\right)^\dagger L^\ell_{k+1}\right) r_0 \\ &= \left(I_n - AL_k\left(H^h_{k+1,k}\right)^\dagger L^\ell_{k+1}\right) r^{Hess}_k. \end{aligned}$$

### 3.2.5 Lanczos biorthogonalization algorithm and the BiCG/QMR pair

We choose $\tilde{r}_0$ a vector of $\mathbb{R}^n$ and set $y_i = A^{T^{i-1}}\tilde{r}_0$. With this choice the Generalized Hessenberg algorithm reduces to the Lanczos bi-orthogonalization algorithm, which is a particular case of the two sided Gram-Schmidt algorithm introduced by Parlett [78]. We construct two basis $\{v_1,\ldots,v_k\} = K_k(A,r_0)$ and $\{\tilde{v}_1,\ldots,\tilde{v}_k\} = K_k(A^T,\tilde{r}_0)$ with $r_0 = \delta_1 v_1$ and, for $k \geq 1$, we define $V_k \equiv [v_1,\ldots,v_k]$, $\tilde{V}_k \equiv [\tilde{v}_1,\ldots,\tilde{v}_k]$, and $v_{k+1}$ and $\tilde{v}_{k+1}$ by

1. $\delta_{k+1}v_{k+1} = c_k = \left(I - K_kK^\ell_k\right)A^{k-1}r_0 = \left(I - V_k(Y_k^TV_k)^{-1}Y_k^T\right)Av_k = \left(I - V_kV^\ell_k\right)Av_k$,
2. $\beta_{k+1}\tilde{v}_{k+1} = \tilde{c}_k = \left(I - Y_k(V_k^TY_k)^{-1}V_k^T\right)\left(A^T\right)^{k-1}\tilde{r}_0 = \left(I - \tilde{V}_k(V_k^T\tilde{V}_k)^{-1}V_k^T\right)A^T\tilde{v}_k$, where $Y_k = \left[A^T\tilde{r}_0,\ldots,(A^{k-1})^T\tilde{r}_0\right]$.
3. If we assume that $\tilde{c}_k^Tc_k \neq 0$, which guarantees that the algorithm does not breakdown, $\beta_k$ and $\gamma_k$ at iteration $k$, can be chosen such that $\tilde{v}_k^Tv_k = 1$. Thus $\tilde{V}_k^TV_k = I_k, V^\ell_k = \tilde{V}_k^T$

$$AV_k = V_{k+1}H^{(1)}_{k+1,k} \quad \text{and} \quad A^T\tilde{V}_k = \tilde{V}_{k+1}H^{(2)}_{k+1,k},$$

where

$$H^{(1)}_{k+1,k} = \begin{bmatrix} \tilde{V}_k^TAV_k \\ \\ \delta_{k+1}\left(e^{(k)}_k\right)^T \end{bmatrix} = \begin{bmatrix} H^{(1)}_k \\ \\ \delta_{k+1}\left(e^{(k)}_k\right)^T \end{bmatrix},\ H^{(2)}_{k+1,k} = \begin{bmatrix} V_k^TA^T\tilde{V}_k \\ \\ \beta_{k+1}\left(e^{(k)}_k\right)^T \end{bmatrix} = \begin{bmatrix} H^{(2)}_k \\ \\ \beta_{k+1}\left(e^{(k)}_k\right)^T \end{bmatrix}.$$

Consequently the matrices $H^{(1)}_k$ and $H^{(2)}_k$ are upper Hessenberg matrices, with $H^{(2)}_k = \left(H^{(1)}_k\right)^T$. Therefore the matrix $H^{(1)}_k$ is tridiagonal and will be denoted by $T_k \equiv H^{(1)}_k$. We

also set $T_{k+1,k} \equiv H^{(1)}_{k+1,k}$. Therefore

$$AV_k = V_{k+1}T_{k+1,k} = V_kT_k + \delta_{k+1}v_{k+1}\left(e_k^{(k)}\right)^T \quad \text{and} \quad A^T\tilde{V}_k = \tilde{V}_kT_k^T + \beta_{k+1}v_{k+1}\left(e_k^{(k)}\right)^T$$

The Lanczos bi-orthogonalization algorithm generates two rectangular matrices $V_k$, $\tilde{V}_k$ and a tridiagonal matrix $T_k$,

$$T_k = \begin{pmatrix} \alpha_1 & \beta_2 & & & \\ \delta_2 & \alpha_2 & \beta_3 & & \\ & \ddots & \ddots & \ddots & \\ & & \delta_{k-1} & \alpha_{k-1} & \beta_k \\ & & & \delta_k & \alpha_k \end{pmatrix}.$$

It is a Krylov method defined by the Petrov-Galerkin condition $r_k^B \perp K_k(A^T, \tilde{r}_0)$. Hence

$$\begin{aligned} r_k^B &= \left(I_n - AV_k(\tilde{V}_k^TAV_k)^{-1}\tilde{V}_k^T\right)r_0 \\ &= r_0 - \|r_0\|V_{k+1}T_{k+1,k}\left(T_k\right)^{-1}e_1^{(k)}, \end{aligned}$$

and its iterates are defined by $x_k^B = x_0 + B_ky_k^B$. The coefficients $y_k$ are computed by requiring orthogonality of the residuals. They are obtained by solving

$$T_k\, y_k^B = \|r_0\|e_1.$$

Hence the involved left inverse of $AV_k$ is $(AV_k)^\ell = T_k^{-1}\tilde{V}_k^T$ and $T_k^{-1}\tilde{V}_k^Tr_0 = \|r_0\|\ T_k^{-1}e_1 = \delta_1\ \ T_k^{-1}e_1$. The following lemma will be used to show that, for computing the iterates $x_k^L$ iteratively, we need only the last row and the last column of $T_k^{-1}$.

**Lemma 3.8.** *If the matrices $T_{k-1}$ and $T_k$ are invertible and we set $s_k = T_k^{-1}e_k$ and $q_k = T_k^{-T}e_k$, then*

1. *the inverse of the tridiagonal matrix $T_k$ is given by*

$$T_k^{-1} = \begin{bmatrix} T_{k-1}^{-1} + \frac{\beta_k \delta_k}{\theta_k} s_{k-1} q_{k-1}^T & -\frac{\beta_k}{\theta_k} s_{k-1} \\ -\frac{\delta_k}{\theta_k} q_{k-1}^T & \frac{1}{\theta_k} \end{bmatrix},$$

*where* $\theta_k = \alpha_k - \beta_k \delta_k \, q_{k-1}^T e_{k-1}$,

2. $s_k = \begin{bmatrix} -\frac{\beta_k}{\theta_k} s_{k-1} & \frac{1}{\theta_k} \end{bmatrix}^T$, *and* $q_k = \begin{bmatrix} -\frac{\delta_k}{\theta_k} q_{k-1} & \frac{1}{\theta_k} \end{bmatrix}^T$, *for* $k \geq 2$ *and* $s_1 = q_1 = \frac{1}{\alpha_1}$,

3. $\theta_k = \frac{1}{q_k^T e_k} = \alpha_k - \frac{\beta_k \delta_k}{\theta_{k-1}}$ *for* $k \geq 2$ *and* $\theta_1 = \alpha_1$,

4. $(q_k, e_1^{(k)}) = -\frac{\delta_k}{\theta_k}(q_{k-1}, e_1^{(k-1)})$, $(s_k, e_1^{(k)}) = -\frac{\beta_k}{\theta_k}(s_{k-1}, e_1^{(k-1)})$ *for* $k \geq 2$.

*Proof.* This is straightforward from the fact that $T_k$ can be written as

$$T_k = \begin{bmatrix} T_{k-1} & \beta_k e_{k-1} \\ \delta_k e_{k-1}^T & \alpha_k \end{bmatrix}.$$

□

Let $p_k$ be the vector $p_k = V_k T_k^{-1} e_k$, using Lemma 3.8, we deduce that the $k$th iterates $x_k^B$ can be written, for $k \geq 2$, as

$$\begin{aligned} x_k^B &= x_{k-1}^B - \|r_0\| \, \delta_k \, (q_{k-1}, e_1^{(k-1)}) \, p_k \\ &= x_{k-1}^B + \|r_0\| \, \theta_k \, (q_k, e_1^{(k)}) \, p_k \qquad \text{and} \quad p_k = \frac{1}{\theta_k}(v_k - \beta_k p_{k-1}), \end{aligned}$$

with $p_1 = \frac{1}{\alpha_1} v_1$. This gives us the direct version of Lanczos bi-orthogonalization algorithm.

The classical implementation of the algorithm, is usually deduced by using the LDU decomposition of $T_k$; see for example [84]. We can deduce it also from our approach by setting

$$\tilde{p}_k = \tilde{V}_k \, (T_k)^{-T} \, e_k^{(k)}.$$

**Algorithm 1** Direct Lanczos Biorthogonalization algorithm [89, 105]

Choose $x_0$, compute $r_0 = f - Ax_0$, $\delta_1 = \|r_0\|$ and choose $\tilde{r_0}$ such that $(\tilde{r_0}, r_0) \neq 0$,
set $v_0 = \tilde{v}_0 = p_0 = 0$, $v_1 = r_0/\delta_1$, $\beta_1 = (\tilde{r_0}, v_1)$, $\tilde{v}_1 = \tilde{r_0}/\beta_1$, $\zeta_1 = \delta_1$ and $\lambda_1 = 0$
for $k = 1, \cdots,$, until convergence, Do:

Compute $u := Av_k - \beta_k v_{k-1}$, $\tilde{u} := A^T\tilde{v}_k - \delta_k\tilde{v}_{k-1}$ and $\alpha_k = (\tilde{u}, v_k)$

If $k > 1$ then compute $\lambda_k = \dfrac{\delta_k}{\theta_{k-1}}$ and $\zeta_k = -\lambda_k\zeta_{k-1}$

$\theta_k = \alpha_k - \lambda_k\beta_k$

$p_k = \dfrac{1}{\theta_k}(v_k - \beta_k p_{k-1})$

$x_k = x_{k-1} + \zeta_k p_k$,

If $x_k$ has converged then Stop,

$u := u - \alpha_k v_k$ and $\tilde{u} := \tilde{u} - \alpha_k\tilde{v}_k$

$\delta_{k+1} = \sqrt{|(\tilde{u}, u)|}$ and $v_{k+1} = u/\delta_{k+1}$

$\beta_{k+1} = (\tilde{u}, b_{k+1})$ and $\tilde{v}_{k+1} = \tilde{u}/\beta_{k+1}$

EndDo.

If we set $P_k = [p_1, \ldots, p_k]$ and $\tilde{P}_k = [\tilde{p}_1, \ldots, \tilde{p}_k]$, we obtain the following properties

1. $span\{p_1, \ldots, p_k\} = span\{v_1, \ldots, v_k\} = K_k(A, r_0)$,
2. $span\{\tilde{p}_1, \ldots, \tilde{p}_k\} = span\{\tilde{v}_1, \ldots, \tilde{v}_k\} = K_k(A^T, \tilde{r}_0)$,
3. the matrix $\tilde{P_k}^T A P_k$ is diagonal, with $\tilde{p_k}^T A p_k = (q_k, e_k) = \dfrac{1}{\theta_k}$.

Using these new bases, we have

$$\begin{aligned} x_k^B &= x_0 + P_k(\tilde{P}_k^T A P_k)^{-1}\tilde{P}_k^T r_0 \\ &= x_0 + \|r_0\|\sum_{i=1}^{k}\frac{(q_i, e_1^{(i)})}{(\tilde{p}_i, Ap_i)}\ p_i \\ &= x_{k-1}^B + \|r_0\|\frac{(q_k, e_1^{(k)})}{(\tilde{p}_k, Ap_k)}\ p_k \qquad \text{and} \quad r_k^B = r_{k-1}^B - \|r_0\|\frac{(q_k, e_1^{(k)})}{(\tilde{p}_k, Ap_k)}\ Ap_k. \end{aligned}$$

On the other hands, from Theorem 3.6, we know that $r_k^B = -\|r_0\|\delta_{k+1}(q_k, e_1)v_{k+1}$, it follows that

$$\begin{aligned} p_k &= \frac{1}{\theta_k}v_k - \frac{\beta_k}{\theta_k}p_{k-1} \\ &= -\frac{1}{\theta_k\|r_0\|\delta_k(q_{k-1}, e_1)}r_{k-1}^B - \frac{\beta_k}{\theta_k}p_{k-1}. \end{aligned}$$

Therefore, using the last statement of Lemma 3.8, we obtain

$$\|r_0\|\theta_k^2(q_k, e_1)p_k = r_{k-1}^B + \|r_0\|(q_{k-1}, e_1)\beta_k\delta_k p_{k-1}.$$

Replacing $k$ by $k+1$ in the last formula and setting $p_k^B = \|r_0\|\theta_{k+1}^2(q_{k+1}, e_1)p_{k+1}$ we obtain $p_0^B = r_0$, $\theta_1 = \alpha_1$ and

$$\begin{aligned} p_k^B &= r_k^B + \frac{\beta_{k+1}\delta_{k+1}}{\theta_k^2} p_{k-1}^B, \qquad x_k^B = x_{k-1}^B + \frac{1}{\theta_k} p_{k-1}^B \\ r_k^B &= r_{k-1}^B - \frac{1}{\theta_k} A p_{k-1}^B, \qquad \theta_k = \alpha_k - \frac{\beta_k \delta_k}{\theta_{k-1}}. \end{aligned}$$

Define similarly the vectors $\tilde{r}_0 = \gamma_1 \tilde{v}_1$, $\tilde{r}_k$ and $\tilde{p}_k^Q$ by

$$\tilde{r}_k = \tilde{r}_0 - A^T \tilde{V}_k (T_k^T)^{-1} V_k^T \tilde{r}_0 \quad \text{and} \quad \tilde{p}_k^B = \gamma_1 \theta_{k+1}^2 (s_{k+1}, e_1)\tilde{p}_{k+1},$$

we obtain

$$\tilde{p}_k^B = \tilde{r}_k^B + \frac{\beta_{k+1}\delta_{k+1}}{\theta_k^2}\tilde{p}_{k-1}^B \quad \text{and} \quad \tilde{r}_k^B = \tilde{r}_{k-1}^B - \frac{1}{\theta_k} A^T \tilde{p}_{k-1}^B.$$

Moreover using the fact that $\tilde{r}_k^B = \gamma_1 \beta_{k+1}(s_k, e_1)\tilde{v}_{k+1}$ and Lemma 3.8, we get

$$(\tilde{r}_{k-1}^B, r_{k-1}^B) = \gamma_1 \|r_0| \theta_k^3 (s_k, e_1)(q_k, e_1) \quad \text{and} \quad (\tilde{p}_{k-1}^B, A p_{k-1}^B) = \gamma_1 \|r_0| \beta_{k+1}\delta_{k+1}(s_k, e_1)(q_k, e_1).$$

Consequently

$$\frac{1}{\theta_k} = \frac{(\tilde{r}_{k-1}^B, r_{k-1}^B)}{(\tilde{p}_{k-1}^B, A p_{k-1}^B)} \quad \text{and} \quad \frac{\beta_{k+1}\delta_{k+1}}{\theta_k^2} = \frac{(\tilde{r}_k^B, r_k^B)}{(\tilde{r}_{k-1}^B, r_{k-1}^B)}.$$

Thus

$$\tilde{p}_k^B = \tilde{r}_k^B + \frac{(\tilde{r}_k^B, r_k^B)}{(\tilde{r}_{k-1}^B, r_{k-1}^B)}\tilde{p}_{k-1}^B \quad \text{and} \quad \tilde{r}_k^B = \tilde{r}_{k-1}^B - \frac{(\tilde{r}_{k-1}^B, r_{k-1}^B)}{(\tilde{p}_{k-1}^B, A p_{k-1}^B)} A^T \tilde{p}_{k-1}^B.$$

Since BiCG ia a Petrov-Galerkin method, its corresponding Minimizing residual semi-norm is the QMR method. Then if we denote by $r_k^Q$ the $k$th residual of the QMR method, we deduce from (3.12) that

$$r_k^Q = r_0 - \delta_1 \, V_{k+1} T_{k+1,k} \, (T_{k+1,k})^{\dagger} \, e_1^{(k+1)}.$$

**Algorithm 2** BiCG method [37, 89]

Choose $x_0$, $r_0 = f - Ax_0$, choose $\tilde{r}_0$ such that $(\tilde{r}_0, r_0) \neq 0$,
set $p_0 = r_0$, and $\tilde{p}_0 = \tilde{r}_0$,
for $k = 0, 1, \cdots$ ,, until convergence Do:

$\alpha_k = (\tilde{r}_k, r_k) \,/\, (\tilde{p}_k, Ap_k)$
$x_{k+1} = x_k + \alpha_k p_k$
$r_{k+1} = r_k - \alpha_k A p_k$
$\tilde{r}_{k+1} = \tilde{r}_k - \alpha_k A^T \tilde{p}_k$
$\beta_k = (\tilde{r}_{k+1}, r_{k+1}) \,/\, (\tilde{r}_k, r_k)$
$p_{k+1} = r_{k+1} + \beta_k p_k$
$\tilde{p}_{k+1} = \tilde{r}_{k+1} + \beta_k \tilde{r}_k$

EndDo.

We have the following Petrov-Galerkin orthogonality conditions

$$\tilde{V}_k^T r_k^B = 0, \qquad \text{and} \qquad T_{k+1,k}^T \tilde{V}_{k+1}^T r_k^Q = 0.$$

Moreover from (3.13) we know that $T_{k+1,k}^{\dagger} \tilde{V}_{k+1}^T$ is a left inverse of $AV_k = T_{k+1,k} V_{k+1}$, and we have

$$\begin{aligned} r_k^Q &= \left(I_n - AV_k \left(T_{k+1,k}\right)^{\dagger} \tilde{V}_{k+1}^T\right) r_0 \\ &= \left(I_n - AV_k \left(T_{k+1,k}\right)^{\dagger} \tilde{V}_{k+1}^T\right) r_k^{Bcg}. \end{aligned}$$

By using the second statement of Theorem 3.5, we obtain

$$\begin{aligned} x_k^Q &= x_{k-1}^Q + \frac{\|r_0\|(q_k, e_1^{(k)})}{(q_k, q_k)\left(1 + \delta_{k+1}^2 (q_k, q_k)\right)} V_k T_k^{-1} q_k \\ &= x_{k-1}^Q + \frac{\|r_0\|(q_k, e_1^{(k)})}{(q_k, q_k)\left(1 + \delta_{k+1}^2 (q_k, q_k)\right)} p_k^Q \qquad \text{with} \quad p_k^Q = V_k T_k^{-1} q_k, \end{aligned}$$

From Theorem 3.7, we de decuce that

$$q_k^T = \frac{1}{\theta_k}\begin{bmatrix} -\delta_k q_{k-1}^T & 1 \end{bmatrix} \quad \text{and} \quad (q_k, q_k) = \frac{1 + \delta_k^2 (q_{k-1}, q_{k-1})}{\theta_k^2}.$$

It follows that

$$\begin{aligned} p_k^Q &= V_k T_k^{-1} q_k = [V_{k-1}, v_k] \begin{bmatrix} -\dfrac{\delta_k}{\theta_k} T_{k-1}^{-1} q_{k-1} - \dfrac{\beta_k (1 + \delta_k^2 (q_{k-1}, q_{k-1}))}{\theta_k^2} s_{k-1} \\ \dfrac{1 + \delta_k^2 (q_{k-1}, q_{k-1})}{\theta_k^2} \end{bmatrix} \\ &= -\frac{\delta_k}{\theta_k} p_{k-1}^Q + \theta_k (q_k, q_k) p_k. \end{aligned}$$

**Algorithm 3** Quasi-Minimal Residual Method (QMR)

Choose $x_0$, compute $r_0 = f - Ax_0$, $\delta_1 = \|r_0\|$ and choose $\tilde{r_0}$ such that $(\tilde{r_0}, r_0) \neq 0$, set $v_0 = \tilde{v}_0 = p_0 = 0$, $v_1 = r_0/\delta_1$, $\beta_1 = (\tilde{r_0}, v_1)$, $\tilde{v}_1 = \tilde{r_0}/\beta_1$, $\zeta_1 = \delta_1$, $\lambda_1 = 0$ and $\rho_0 = 0$
for $k = 1, \cdots,$, until convergence Do:

Compute $u := Av_k - \beta_k v_{k-1}$, $\tilde{u} := A^T\tilde{v}_k - \delta_k\tilde{v}_{k-1}$ and $\alpha_k = (\tilde{u}, v_k)$

If $k > 1$ then compute $\lambda_k = \dfrac{\delta_k}{\theta_{k-1}}$ and $\zeta_k = -\lambda_k\zeta_{k-1}$

$\theta_k = \alpha_k - \lambda_k\beta_k$

$p_k = \dfrac{1}{\theta_k}(v_k - \beta_k p_{k-1})$

$p_k^Q = -\dfrac{\delta_k}{\theta_k}p_{k-1}^Q + \theta_k\rho_k p_k$

$u := u - \alpha_k v_k$ and $\tilde{u} := \tilde{u} - \alpha_k\tilde{v}_k$

$\delta_{k+1} = \sqrt{|(\tilde{u}, u)|}$ and $v_{k+1} = u/\delta_{k+1}$

$\beta_{k+1} = (\tilde{u}, v_{k+1})$ and $\tilde{v}_{k+1} = \tilde{u}/\beta_{k+1}$

$\rho_k = \dfrac{(1 + \delta_k^2\rho_{k-1})}{\theta_k^2}$

$x_k = x_{k-1} + \dfrac{\zeta_k\theta_k}{\rho_k(1 + \delta_{k+1}^2\rho_k)}p_k^Q,$

If $x_k$ has converged then Stop,

EndDo.

## 3.3 Enhancement of the convergence behaviour of some classical Krylov methods

In this section, we first discuss the choice of the $Z_k$ matrix. Then, we will discuss a strategy for improving the convergence of BiCG and BiCGstab. A similar approach for improving BiCGSTAB and IDR, which are product-type Krylov subspace methods, can be found in [22, 23]. The $Z_k$ matrix should be chosen in such a way as to limit work and storage per iteration. The improved algorithm should involve a limited number of vectors and very little extra work. The preferred algorithms are those for which this number is low (less than ten). In addition, we need to avoid adding additional matrix-vector products.

By invoking Theorem 3.2 with $r_k^P$ a residual vector of a Petrov-Galerkin method, we obtain

$$r_k^K = (I_n - AK_kZ_k^T)r_k^P, \qquad \text{and} \qquad x_k^K = x_0 + K_kZ_k^Tx_k^P. \tag{3.17}$$

The matrix $Z_k$ will be chosen such that $\|(I_k - W_k Z_k^T)\| \leq 1$. Then, using the preceding relation, we get

$$\|r_k^K\| = \|(I_n - W_k Z_k^T) r_k^P\| \leq \|r_k^P\|.$$

Let us consider in the following, some applications.

### 3.3.1 Enhanced BiCG method

As we have already seen, in BiCG, at each iteration of Krylov subspace methods, we compute the vectors $p_k$ and $Ap_k$, such that

$$span\{p_0, p_1, \ldots, p_{k-1}\} = K_k(A, r_0).$$

Let $P_{k,s_k}$ be the matrix defined by $P_{k,s_k} = [p_{k-s_k}, \ldots, p_{k-1}]$ with $1 \leq s_k \leq k$. The matrix $Z_k$ can be chosen such that

$$I - W_k Z_k^T = I - AP_{k,s_k} \left(AP_{k,s_k}\right)^{\dagger}.$$

Then we obtain a new residual vectors such that $\|r_k^{EB}\| \leq \|r_k^B\|$, with

$$r_k^{EB} = (I - (AP_{k,s_k})((AP_{k,s_k})^{\dagger}) r_k^B. \tag{3.18}$$

If $s_k = k$,we have to use all the preceding vectors $p_0, \ldots, p_{k-1}$. We propose to use only the a fixed small number of vectors. Hence we optimize the cost of work and the memory per iteration of the modified algorithm. For example if we $s_k = 1$ then we obtain the simplest Enhanced BiCG algorithm. In this case we will refer to this algorithm as EBiCG(1)

$$r_k^{EB} = r_k^B - Ap_{k-1} \left(Ap_{k-1}\right)^{\dagger} r_k^B, \tag{3.19}$$

and

$$x_k^B = x_k^{\mathrm{B}} + p_{k-1} \left(Ap_{k-1}\right)^{\dagger} r_k^{EB}. \tag{3.20}$$

We consider these new vectors, we get Simplest Enhanced BiCG method.

**Algorithm 4** Simplest Enhanced BiCG method ($s_k = 1, \forall k$)

Choose $x_0$, $r_0 = f - Ax_0$, choose $\tilde{r}_0$ such that $(\tilde{r}_0, r_0) \neq 0$,
set $p_0 = r_0$, and $\tilde{p}_0 = \tilde{r}_0$,
for $k = 0, 1, \cdots$ ,, until convergence Do:
$\alpha_k = (\tilde{r}_k, r_k) \,/\, (\tilde{p}_k, Ap_k)$
$x_{k+1} = x_k + \alpha_k p_k$
$u_k = Ap_k$
$r_{k+1} = r_k - \alpha_k u_k$
$\rho_k = (u_k, r_{k+1}) \,/\, (u_k, u_k)$
$x^E_{k+1} = x_{k+1} + \rho_k p_k$
$r^E_{k+1} = r_{k+1} - \rho_k u_k$
$\tilde{r}_{k+1} = \tilde{r}_k - \alpha_k A^T \tilde{p}_k$
$\beta_k = (\tilde{r}_{k+1}, r_{k+1}) \,/\, (\tilde{r}_k, r_k)$
$p_{k+1} = r_{k+1} + \beta_k p_k$
$\tilde{p}_{k+1} = \tilde{r}_{k+1} + \beta_k \tilde{r}_k$
EndDo.

### 3.3.2 Some enhanced Krylov subspaces product type methods

The Krylov methods considered in this section are products methods in which the *kth* residual is expressed by $r_k = \Psi_k(A)\Phi_k(A)r_0$, where $\Phi_k$ is a polynomial of degree $k$ such that $\Phi_m$ is the minimal polynomial of $A$ for $r_0$ and $\Psi_k$ a polynomial of degree $k$. Sonneveld [83] propose to modify the BiCG by replacing the multiplication by $A^T$ by a second one with $A$. The residual is CGS is $r_k^{CGS} = \Phi_k^2(A)r_0$, which correspond to $\Psi_k = \Phi_k$. If we set the direction vector $p_k^B$ in terms of polynomial in A applied to the initial residuals, as

$$r_k^{BiCG} = \Phi_k(A)r_0, \quad p_k^{CGS} = \Theta_{k-1}^2(A)r_0,$$

where $\Theta_k$ is polynomial of degree less than or equal to $k$, we can derive the CGS algorithm by setting

$$r_k^{CGS} = \Phi_k^2(A)r_0, \quad p_k^{CGS} = \Theta_{k-1}^2(A)r_0, \quad u_k = \Phi_{k-1}(A)\Theta_{k-1}(A)r_0, \quad \text{and} \quad q_k = \Phi_k(A)\Theta_{k-1}(A)r_0.$$

To avoid increasing the cost at each iteration, we can use the vectors already computed and those with a matrix vector multiplication already done. For the storage problems,

**Algorithm 5** CGS method [83]

Choose $x_0$, $r_0 = f - Ax_0$, choose $\tilde{r}_0$ such that $(\tilde{r}_0, r_0) \neq 0$,
set $p_0 = u_0 = r_0$,
for $k = 0, 1, \cdots$ ,, until convergence Do:
$\alpha_k = (\tilde{r}_0, r_k) \,/\, (\tilde{r}_0, Ap_k)$
$q_k = u_k - \alpha_k A p_k$
$x_{k+1} = x_k + \alpha_k (u_k + q_k)$
$r_{k+1} = r_k - \alpha_k A (u_k + q_k)$
$\beta_k = (\tilde{r}_0, r_{k+1}) \,/\, (\tilde{r}_0, r_k)$
$u_{k+1} = r_{k+1} + \beta_k q_k$,
$p_{k+1} = u_{k+1} + \beta_k (q_k + \beta_k p_k)$,
EndDo.

we will use the vectors $p_k$ and $u_k + q_k$ computed at each iteration. So, concerning the CGS method, we remark that at each iteration we calculate two vectors $Ap_k$ and $A(u_k + q_k)$ and we will use them to build the orthogonal projector. We get

$$r_{k+1}^{\mathrm{ECGS}} = r_{k+1}^{\mathrm{CGS}} - [Ap_k, A(u_k + q_k)] \left[Ap_k, A(u_k + q_k)\right]^{\dagger} r_{k+1}^{\mathrm{CGS}}, \tag{3.21}$$

and

$$x_{k+1}^{\mathrm{ECGS}} = x_{k+1}^{\mathrm{CGS}} + [p_k, (u_k + q_k)] \left[Ap_k, A(u_k + q_k)\right]^{\dagger} r_{k+1}^{CGS}. \tag{3.22}$$

Of course we can use more than the last two vectors. A natural generalization is to store a fixed number of preceding vectors from the sequence $\{p_k, u_k + q_k\}$, that is,

$$r_{k+1}^{ECGS} = (I - (AP_{k,s_k})((AP_{k,s_k})^{\dagger}) r_{k+1}^{CGS}, \tag{3.23}$$

where

$$P_{k,s_k} = [p_{k-s_k}, u_{k-s_k} + q_{k-s_k} \ldots, p_{k-1}, u_k + q_k].$$

In [22, 23], similar techniques was used to improve the convergence behaviour of BiCGSTAB and IDR.

## 3.4 Numerical examples

In this section, we consider the following convection-diffusion equation

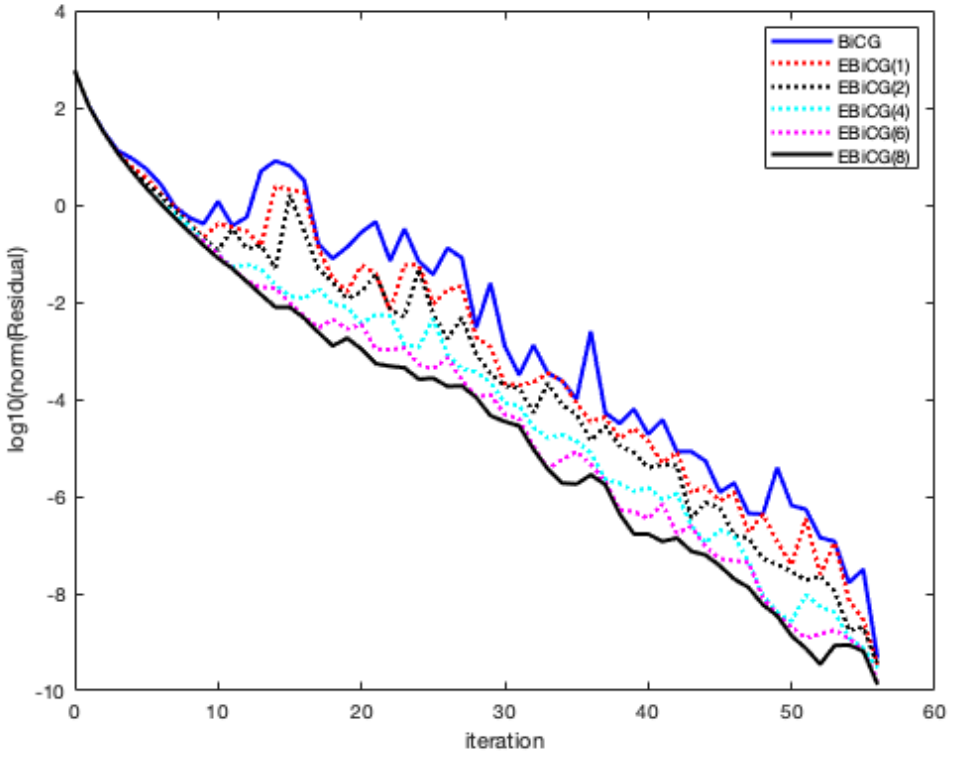


**Figure 3.1:** *Comparison between residual norms of BiCG and its enhancements EBiCG(s), for* $s = 1, 2, 4$.

$$\begin{cases} -\Delta u - \alpha.\nabla u - \beta u = f, & \text{in } \Omega \\ u = 0, & \text{on } \partial\Omega, \end{cases}$$

where $\Omega = [0,1]^3$ and $\alpha = (\alpha_x, \alpha_y, \alpha_z)^T \in \mathbb{R}^3$. The discretization of this equation is performed using centered finite differences with the standard 7-point stencil in three dimensions, resulting in a sparse matrix. For all examples, we choose $\alpha = (0.5, 0.5, 0.5)^T$, $\beta = 5$, and $N_x = 30, N_y = 20, N_z = 20$. The dimension of the system is $N = N_x \times N_y \times N_z = 12\,000$.

In these numerical examples, the right-hand side $b$ of the system is determined by

$$x^\star = (1, \ldots, 1)^T, \quad \text{and} \quad b = Ax^\star,$$

where $x^\star$ represents the exact solution of the given system. The initial guess is set to zero. The tests are terminated as soon as $|r_k|/|b| \leq 10^{-10}$.

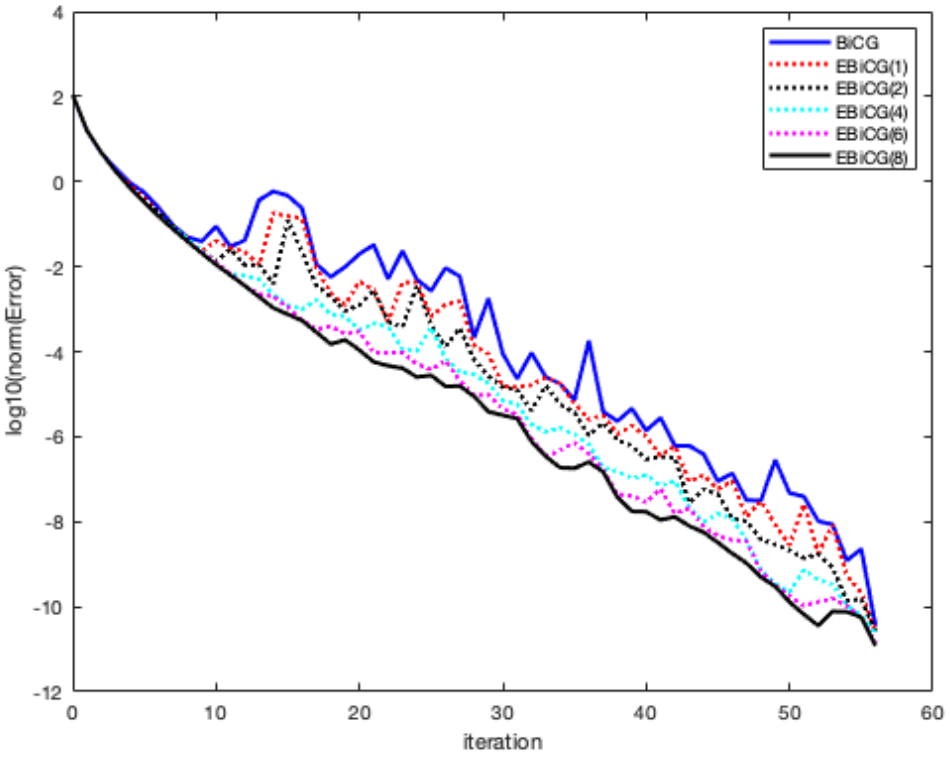


**Figure 3.2:** *The comparison between error norms of BiCG and its enhancements EBiCG(s), for* $s = 1, 2, 4$.

To demonstrate the effectiveness of our technique, we compare the BiCG method with its enhanced variants, as outlined in Algorithm 4, utilizing only the previously stored direction vectors $p_{k-s+1}, \ldots, p_k$. The simplest enhanced BiCG algorithm corresponds to the case where $s = 1$. Additionally, we compare the cases where $s = 2, 4, 8$, which involve the utilization of formula (3.18) with 2, 4, or 8 direction vectors. We denote the Enhanced BiCG algorithm with $s$ direction vectors $p_{k-s+1}, \ldots, p_k$ as EBiCG($s$). To illustrate the comparison of these algorithms in terms of residual and error norms, we present the curves of residual norms and error norms for $s = 1, 2, 4, 8$ in Fig. 3.1 and Fig. 3.2, respectively.

Fig. 3.3 and Fig. 3.4 display the curves depicting the residual norms and error norms for the CGS algorithm and its enhancements. In particular, ECGS($s$) refers to the utilization

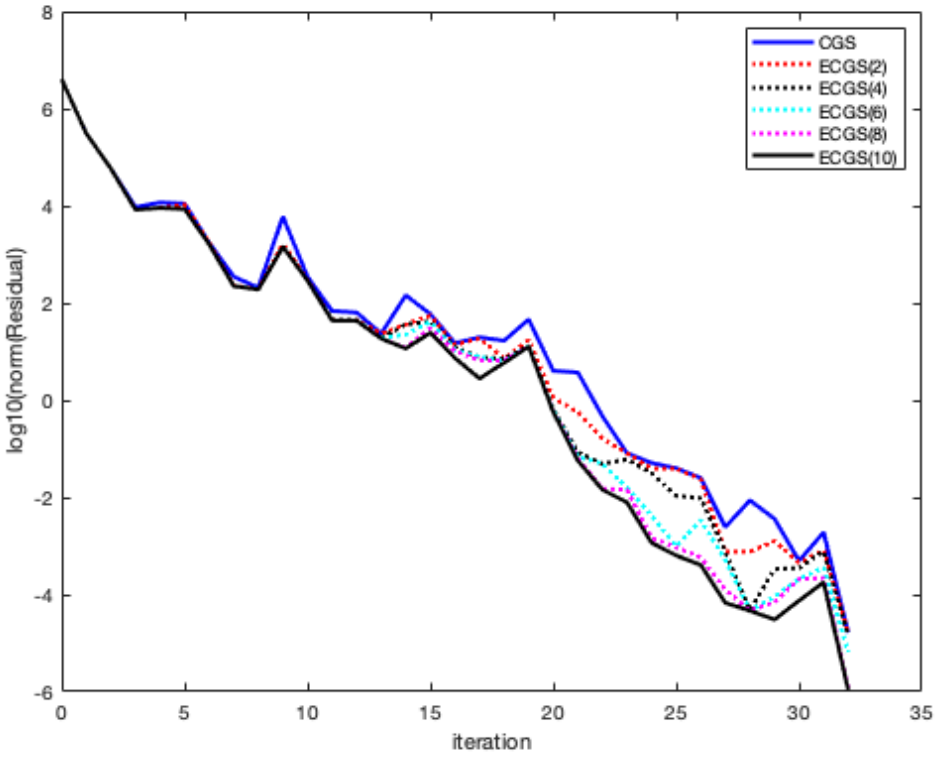


**Figure 3.3:** *The comparison between residual norms of CGS and its enhancements ECGS(s), for* $s = 2, 4, 6, 8$

of formula 3.23, incorporating the available direction vectors $p_{k-i}, u_{k-i}$ for $i = 1, \ldots, s$.

## 3.5 Conclusion

In this paper we have presented a comprehensive framework for studying Krylov subspace methods, explored their mathematical properties and convergence behaviour, and discussed techniques to enhance their performance. The paper covers various aspects including the minimal polynomial of matrix $A$, the relationship between different methods, the role of generalized inverses, and the use of product methods. The provided numerical examples further support the analysis and conclusions of the research. The paper introduces a comprehensive framework for studying Krylov subspace methods used to solve linear systems of the form $Ax = f$, where $A$ is a matrix, $x$ is the unknown vector, and $f$ is the right-hand side vector. The objective of these methods is to achieve convergence within a specified number of iterations, denoted as $m$.

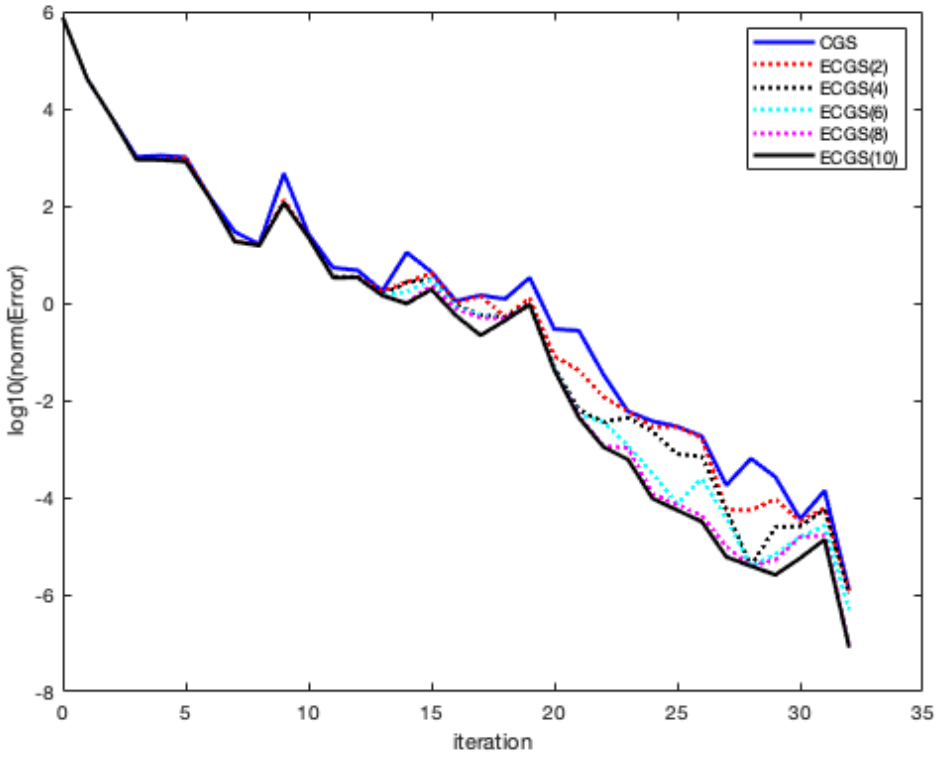


**Figure 3.4:** *The comparison between error norms of CGS and its enhancements ECGS(s), for* $s = 2, 4, 6, 8$

The minimal polynomial $\Phi_m$ of matrix $A$ for the initial residual $r_0 = f - Ax_0$ is a key focus of the analysis in the paper. The degree of $\Phi_m$ is $m$, and the properties of this minimal polynomial play a crucial role in the convergence behaviour of the Krylov subspace methods.

The paper establishes that Petrov-Galerkin methods and minimal seminorm methods are specific cases of the broader framework of Krylov subspace methods. Additionally, it is demonstrated that minimal seminorm methods satisfy implicit Petrov-Galerkin conditions.

The paper presents a general formulation for the iterates of Krylov subspace methods based on generalized inverses. The choice of a specific left inverse and the construction method of the Krylov basis are important factors that differentiate various Krylov subspace methods. The mathematical properties of these methods are described and analysed, with emphasis on their dependency on two matrices.

The paper proves that specific instances of Krylov subspace methods, such as CMRH (Conjugate Minimum Residual with Hessenberg matrix) and QMR (Quasi-Minimal Residual), satisfy implicit Petrov-Galerkin orthogonality conditions.

Techniques for improving the convergence behavior of Krylov subspace methods by carefully selecting vectors in their implementations are explored. The aim is to deepen the understanding of these methods, provide insights into their convergence properties, and identify potential enhancements.

The paper also discusses Krylov methods that are product methods, where the $k$th residual $r_k$ associated with the approximation $x_k$ of the exact solution is expressed as $r_k = \Psi_k(A)\Phi_k(A)r_0$. Here, $\Psi_k$ is a polynomial of fixed or variable degree. Specific choices of $\Psi_k$, including local convergence, smoothing, fixed memory, and cost considerations for each iteration, are examined.

Enhancements of product methods such like CGS (Conjugate Gradient Squared) is presented in the paper.

In conclusion, the paper presents a comprehensive framework for studying Krylov subspace methods, investigates their mathematical properties and convergence behaviour, explores techniques for improvement, and provides numerical examples to demonstrate the effectiveness of the proposed algorithms.

# Chapter 4

# An enhancement of the convergence of the BiCGStab method for solving linear systems with single or multipe right hand side


## Abstract

In this paper, we present a technique to improve the convergence of the Bi-Conjugate Gradient Stabilized (BiCGStab) method. This method was developed by Van der Vorst for solving nonsymmetric linear systems with a single right-hand side. The global and block versions of the BiCGStab method have been proposed for solving nonsymmetric linear systems with multiple right-hand sides. Using orthogonal projectors to minimize the residual norm in each step, we get an enhancement of the convergence of each version of the BiCGStab method. The considered methods are BiCGStab, global BiCGStab, and block BiCGStab methods, noted respectively Gl-BiCGStab and Bl-BiCGStab. To show the performance of our enhanced algorithms, we compare them with the standard, global, and block versions of the well-known Generalized Minimal Residual method (GMRES).

## 4.1 Introduction

The aim of the BiCGStab method studied in this paper is to solve the following non-symmetric linear system

$$Ax = b, \tag{4.1}$$

where $A$ is a nonsingular matrix in $\mathbb{R}^{N\times N}$ and the vectors $b$ and $x$ are in $\mathbb{R}^N$. Problems such as (5.1) occur in most applications of scientific computing, engineering applications, and Navier-Stokes equations in computational fluid dynamics and structural mechanics computations based on the finite element analysis. If the order $N$ of the matrix $A$ is small, we can solve (5.1) using direct methods, but if $N$ is large, direct methods can be prohibitively expensive both in terms of memory and time. So, iterative methods become appealing. These methods include Krylov subspace methods. It is true that current methods give us interesting computational performance, but it is important to realize that there is no single method that can solve every linear system with desirable accuracy. However, a number of factors can influence the choice of the method, including the conditioning of the matrix $A$ and the number of nonzero values. In literature, there are many problems of linear systems with a sparse matrix and which are obtained from real applications (heat transfer, fluid flow, mass transport, etc) by using numerical strategy for solving partial differential equations (finite difference, finite volume, and finite element methods). The difference between all the existing methods for solving linear systems is in the level of accuracy, turnaround time, and storage. For system (5.1) with a symmetric positive definite matrix, we use the conjugate gradient method (CG). If $A$ is a symmetric matrix, we use usually the MINRES. Finally, the nonsymmetric case can be solved using GMRES developed as the most popular and the most optimal in terms of precision but suffers from storage problems. The nonsymmetric case can also be solved using some short-recurrence Krylov subsapce methods. We find for example the BiCG, CGS, and BiCGStab methods. These methods are derived from the extension of the CG in the non-symmetric case. In this work, we focus on the non-symmetric case, especially on the BiCGStab method as the most stable method compared with

BiCG and CGS methods. For more details about all Krylov subspace methods see [95] and the current book [73].

The BiCGStab method has been developed in [104] for solving (5.1). The block version of this method has been given in [33] for solving linear systems with several right hand-sides. Using projectors and an idea for improvement the convergence of the IDR method [91], given in [23], we will apply it for giving an enhancement of the BiCGStab method.

The rest of this paper is organized as follows: in the next section, we recall the algorithm of BiCGStab method [104]. Then, we propose an improvement of the convergence of this algorithm using orthogonal projectors. A partial and full improvement of the BiCGStab method is proposed and will be called PEnha-BiCGStab(k) and FEnha-BiCGStab respectively. In Section 3, we focus on the solution of linear systems with multiple right-hand sides. We will recall the global version of the BiCGStab [57], which will be called global BiCGStab (Gl-BiCGStab) method. We will also propose two improvements of this method, partial and full improvements, which will be called partial and full enhancement of the global BiCGStab and denoted by PEnha-Gl-BiCGStab(k) and FEnha-Gl-BiCGStab, respectively. In section 4, we will recall the block version of the BiCGStab (Bl-BiCGStab) method. We will also propose two improvements of this method, partial and full improvements, which will be called partial and full enhancement of the block BiCGStab and denoted by PEnha-Bl-BiCGStab(k) and FEnha-Bl-BiCGStab, respectively.

In Section 5, we will present some numerical experiments to compare the proposed algorithms with the well-known GMRES [92], the global GMRES (Gl-GMRES) [55] and the block GMRES (Bl-GMRES) [103] methods.

Throughout this article, all vectors and matrices are assumed to be real and the following notation is used. First, $M^T$ represents the transpose of any matrix $M$. For two vectors $x$ and $y$ in $\mathbb{R}^N$, the inner product is $\langle x, y\rangle_2 = x^T y$, with $\|x\|_2 = \sqrt{\langle x, x\rangle_2}$ the Euclidean norm. In the block and global cases, we consider for two matrices $X$ and $Y$ in $\mathbb{R}^{N\times m}$. The inner product is defined by $\langle X, Y\rangle_F = \mathrm{Tr}\left(X^T Y\right)$, where $\mathrm{Tr}\left(Z\right)$ denotes the trace of a square matrix $Z$. Moreover, the associated norm is the Frobenius norm

noted $\|.\|_F$. We denote by $I$ the identity matrix of order $N$.

## 4.2 BiCGStab method and its enhancement

The Bi-Conjugate Gradient Stabilized (BiCGStab) algorithm has been developed for solving nonsymmetric linear system (5.1). This algorithm has been given from the Conjugate Gradient Squarred (CGS) algorithm of Sonneveld [83], which is obtained from the Bi-Conjugate Gradient (BiCG) algorithm (see [95]). This last algorithm was obtained by using the Lanczos bi-orthogonalization (see [95]). All this methods are Krylov subspace methods for solving linear systems. So, in this section, we give some theoretical background and some preliminary results. For an initial guess $x_0$, we associate a residual vector $r_0 = b - Ax_0$.

**Definition 4.1.** We define the Krylov subspace of order $k$ associated to the matrix $A$ and vector $r_0$ by

$$\mathcal{K}_k^s(A, r_0) = \text{Span}\left\{r_0, Ar_0, \ldots, A^{k-1}r_0\right\}.$$

Classical Krylov subspace methods compute the approximate solution $x_k$ and its correspond residual $r_k = f - Ax_k$ such that

$$x_k - x_0 \in K_k(A, r_0), \quad \text{and } r_k = \Phi_k(A)r_0 \quad \text{for } k = 1, \ldots, m,$$

where $\Phi_k$ is a polynomial of degree $k$. Let

$$\psi_m(\xi) = \sigma_0 + \sigma_1\xi + \cdots + \sigma_m\xi^m = \sum_{i=0}^{m} \sigma_i\xi^i, \quad \text{with} \quad \psi_m(0) = 1,$$

be the minimal polynomial of the matrix $A$ with respect to the vector $r_0$, i.e.

$$\psi_m(A)r_0 = \sum_{i=0}^{m} \sigma_i A^i r_0 = 0, \tag{4.2}$$

and

$$m = \min \{k \quad \text{such that} \quad \sum_{i=0}^{k} \sigma_i A^i r_0 = 0, \quad \text{with} \quad \sigma_0 = 1\}.$$

If the considered method converge after $m$ iterations, the polynomial matrix $\Phi_k(A)$ can be written as follow

$$\Phi_k(A) = \phi_k(A)\psi_k(A).$$

The way in which the coefficients of these two polynomials are calculated gives us the different variants of Krylov subspace methods. For example, in GMRES method

$$\phi_k(A) = I \text{ and } \psi_k(A) = I - W_k W_k^{\dagger}$$

with $W_k = \left[Ar_0, A^2 r_0, \ldots, A^k r_0\right]$ and $W_k^{\dagger} = \left(W_k^T W_k\right)^{-1} W_k^T$ is the pseudo-inverse of the matrix $W_k$. The BiCG method is obtained if we consider the following choice

$$\phi_k(A) = I \text{ and } \psi_k(A) = p_k(A) = I - W_k \left(\tilde{W}_k^T W_k\right)^{-1} \tilde{W}_k^T$$

with $\tilde{W} = \left[r_0, A^T r_0, \ldots, \left(A^T\right)^{k-1} r_0\right]$. For more details of this characterization see [23, 95].

*Remark* 4.2. $H_k = I - W_k W_k^{\dagger}$ is an orthogonal projector. Then, the associated residual vector of GMRES method is defined by an orthogonal projector, hence the optimal property of this method.

The CGS method is developed to avoid the calculation of the transpose of the matrix $A$ in the BiCG method, then the residual associated to CGS method is given as follow

$$r_k^{CGS} = p_k^2(A)\, r_0 \tag{4.3}$$

The CGS algorithm is based on squaring the residual polynomial, and, in cases of irregular convergence, this may lead to substantial build-up of rounding errors, or possibly even overflow. The BiCGStab algorithm is a variation of CGS method which was developed to remedy this difficulty. Instead of seeking a method which delivers a residual vector of the form (4.3), BiCGStab produces iterates whose residual vectors are of the

form

$$r_k^{BiCGStab} = \Psi_k(A)\, p_k(A)\, r_0 \tag{4.4}$$

in which, as before, $p_k(A)$ is the residual polynomial associated with the BiCG algorithm and $\Psi_k(A)$ is a new polynomial which is defined recursively at each step with the goal of "stabilizing" or "smoothing" the convergence behavior of the original algorithm. Specifically, $\Psi_k(A)$ is defined by the simple recurrence,

$$\Psi_{k+1}(A) = (I - \omega_k A)\, \Psi_k(A) \tag{4.5}$$

with $\omega_k$ is determined by minimizing the norm of the residual. Based on this fact of minimization and from Remark 4.2, our idea of improving the convergence of the BiCGStab method comes from the fact that we can give another form to these polynomials to improve convergence, using the data that we have and keeping the other properties, namely storage and computation time.

The BiCGStab method is summarized by the following algorithm

**Algorithm 6** Bi-Conjugate Gradient Stabilized (BiCGStab) [104]

1. $x_0 \in \mathbb{R}^N$ guess initial vector;
2. $r_0 = b - Ax_0$, $p_0 = r_0$, $\tilde{r}_0 = r_0$;
3. for $i = 0, 1, 2 \ldots$;
4. $\quad v_i = Ap_i$;
5. $\quad \alpha_i = \langle \tilde{r}_0, r_i \rangle_2 \,/\, \langle \tilde{r}_0, v_i \rangle_2$;
6. $\quad s_i = r_i - \alpha_i v_i$;
7. $\quad t_i = As_i$;
8. $\quad \omega_i = \langle t_i, s_i \rangle_2 \,/\, \langle t_i, t_i \rangle_2$;
9. $\quad x_{i+1} = x_i + \alpha_i p_i + \omega_i s_i$;
10. $\quad r_{i+1} = s_i - \omega_i t_i$;
11. $\quad \beta_i = -\langle \tilde{r}_0, t_i \rangle_2 \,/\, \langle \tilde{r}_0, v_i \rangle_2$;
12. $\quad p_{i+1} = r_{i+1} + \beta_i (p_i - \omega_i v_i)$;
13. end for.

We will propose an improvement of the convergence of the BiCGStab method. Two enhancements of this method are studied, the first one will be called partial enhancement, denoted by PEnha-BiCGStab(k), and the second one will be called full enhancement, denoted by FEnha-BiCGStab. We propose to improve the convergence of the BiCGStab method by using the following well-known result.

**Proposition 4.3.** *Consider the orthogonal projector*

$$Q_l = I - Z_l Z_l^{\dagger},$$

*where the rectangular matrix $Z_l$ is a full rank matrix in $\mathbb{R}^{N\times l}$ and*

$$Z_l^{\dagger} = \left(Z_l^T Z_l\right)^{-1} Z_l^T,$$

*is its pseudo-inverse (Moore-Penrose) (for more details of the pseudo inverse see [13]). Applying the projector $Q_l$ to any vector $r \in \mathbb{R}^N$, we obtain a new vector, which we denote by*

$$r^{Enha} = Q_l r.$$

*Then, we have*

$$\left\| r^{Enha} \right\|_2 \leq \|r\|_2 . \tag{4.6}$$

The proof of this proposition is given in [95] page 38.

To improve the convergence of an iterative method for solving linear systems, it is necessary to minimize and decrease the norm of its residual in as few iterations as possible. Then, by invoking Proposition 5.3 with the residual vector $r_i$, we obtain an improvement in the accuracy and stability of the BiCGStab algorithm. Thus, we will apply an orthogonal projector $Q_k$ to the residual of this method to obtain a new residual with a smaller norm. Furthermore, to avoid a storage problem, we use the $k$ pairs of vectors already calculated in the BiCGStab method to construct the orthogonal projector.

The partial enhancement of the convergence of the BiCGStab method (PEnha-BiCGStab(k))

is given by choosing $Z_k$ equal to the k last pairs of vectors $[v_i, t_i]$, and by adding to line 10 in Algorithm 6 the following instructions

1. $Z1 = [p_{i-k+1}, s_{i-k+1}, p_{i-k+2}, s_{i-k+2}, \ldots, p_i, s_i]$;
2. $Z_k = AZ1 = [v_{i-k+1}, t_{i-k+1}, v_{i-k+2}, t_{i-k+2}, \ldots, v_i, t_i]$;
3. $Z = Z_k^{\dagger} r_i$;
4. $x_i^{PEnha} = x_i + Z1Z$;
5. $r_i^{PEnha} = r_i - Z_k Z$.

The full enhancement of the convergence of the BiCGStab (FEnha-BiCGStab) method is defined by choosing $Z$ equal to the all last pairs of vectors

$$[v_0, t_0, v_1, t_1, \ldots, v_i, t_i],$$

and by adding to line 10 in Algorithm 6 the following instructions

1. $Z1 = [p_0, s_0, p_1, s_1, \ldots, p_i, s_i]$;
2. $Z = AZ1 = [v_0, t_0, v_1, t_1, \ldots, v_i, t_i]$;
3. $Z2 = Z^{\dagger} r_i$;
4. $x_i^{FEnha} = x_i + Z1Z2$;
5. $r_i^{FEnha} = r_i - ZZ2$.

## 4.3 Global BiCGStab method and its enhancement

In this section, we consider the solution of large and sparse nonsymmetric systems with multiple right-hand sides of the form

$$AX = B, \tag{4.7}$$

where the coefficient matrix $A$ is a nonsingular real matrix of order $N$,

$$X = [x_1\ x_2\ \ldots\ x_m] \quad \text{and} \quad B = [b_1\ b_2\ \ldots\ b_m] \in \mathbb{R}^{N\times m},$$

with $m \ll N$.

One class of solvers for solving problem (5.11) are the global methods, which are based on the use of a global projection process onto a matrix (global) Krylov subspace, including global FOM and GMRES methods [55], global BiCG and Bi-CGStab methods [33], global Hessenberg and CMRH methods [32].

The other class is that of the block solvers which are much more efficient when the matrix $A$ is relatively dense. The first block solvers are the block conjugate gradient (Bl-CG) and block bi-conjugate gradient (Bl-BiCG) methods proposed in [74], for non-symmetric problems, the block generalized minimal residual (Bl-GMRES) algorithm [103], the block quasi-minimum residual (Bl-QMR) algorithm [39], the block BiCGStab (Bl-BiCGStab) algorithm [33].

In what follows, we recall the global Bi-Conjugate Gradient Stabilized (Gl-BiCGStab) algorithm.

We will propose an improvement of the convergence of the Gl-BiCGStab method. Two enhancements of this method are studied, the first one will be called partial global enhancement, denoted by PEnha-Gl-BiCGStab(k), and the second one will be called full global enhancement, denoted by FEnha-Gl-BiCGStab. We propose to improve the convergence of the Gl-BiCGStab method by using the following well-known result.

**Proposition 4.4.** *Consider the orthogonal projector*

$$\mathcal{Q}_l = I - \mathcal{Z}_l \mathcal{Z}_l^{\dagger},$$

*where the rectangular matrix $\mathcal{Z}_l$ is a full rank matrix in $\mathbb{R}^{N\times m}$ and*

$$\mathcal{Z}_l^{\dagger} = \left(\mathcal{Z}_l^T \mathcal{Z}_l\right)^{-1} \mathcal{Z}_l^T,$$

**Algorithm 7** Global Bi-Conjugate Gradient Stabilized (Gl-BiCGStab) [57]

1. $X_0 \in \mathbb{R}^{N\times m}$ guess initial matrix;
2. $R_0 = B - AX_0$, $P_0 = R_0$, $\tilde{R}_0 = R_0$;
3. for $i = 0, 1, 2 \ldots$;
4. $V_i = AP_i$;
5. $\alpha_i = \left\langle \tilde{R}_0, R_i \right\rangle_F / \left\langle \tilde{R}_0, V_i \right\rangle_F$;
6. $S_i = R_i - \alpha_i V_i$;
7. $T_i = AS_i$;
8. $\omega_i = \langle T_i, S_i \rangle_F / \langle T_i, T_i \rangle_F$;
9. $X_{i+1} = X_i + \alpha_i P_i + \omega_i S_i$;
10. $R_{i+1} = S_i - \omega_i T_i$;
11. $\beta_i = -\left\langle \tilde{R}_0, T_i \right\rangle_F / \left\langle \tilde{R}_0, V_i \right\rangle_F$;
12. $P_{i+1} = R_{i+1} + \beta_i (P_i - \omega_i V_i)$;
13. end for.

*is its pseudo-inverse (Moore-Penrose). Applying the projector $\mathcal{Q}_l$ to any matrix $R \in \mathbb{R}^{N\times m}$, we obtain a new residual, which we denote by*

$$R^{Enha} = \mathcal{Q}_l R.$$

*Then, we have*

$$\left\| R^{Enha} \right\|_F \leq \|R\|_F . \tag{4.8}$$

The proof of this proposition is similar to that of the proposition in the standard case. By invoking Proposition 5.12 with the residual matrix $R_i$, we obtain an improvement of the convergence Gl-BiCGStab algorithm. Thus, we will apply an orthogonal projector $\mathcal{Q}_k$ to the residual of this method and then to minimize its norm. To avoid a storage problem, we use the $k$ pairs of matrices already calculated in the Gl-BiCGStab method to construct the orthogonal projector.

The partial enhancement of the convergence of the Gl-BiCGStab method (PEnha-Gl-BiCGStab(k)) is given by choosing $\mathcal{Z}_k$ equal to the k last pairs of matrices $[V_i, T_i]$, and

by adding to line 10 in Algorithm 7 the following instructions

1. $\mathcal{Z}1 = [P_{i-k+1}, S_{i-k+1}, P_{i-k+2}, S_{i-k+2}, \ldots, P_i, S_i]$ ;
2. $\mathcal{Z}_k = A\mathcal{Z}1 = [V_{i-k+1}, T_{i-k+1}, V_{i-k+2}, T_{i-k+2}, \ldots, V_i, T_i]$ ;
3. $\mathcal{Z}2 = \mathcal{Z}_k^{\dagger} R_i$;
4. $X_i^{PEnha} = X_i + \mathcal{Z}1\mathcal{Z}2$;
5. $R_i^{PEnha} = R_i - \mathcal{Z}_k\mathcal{Z}2$.

The full enhancement of the convergence of the Gl-BiCGStab (FEnha-Gl-BiCGStab) method is defined by choosing $\mathcal{Z}_l$ equal to the all last pairs of matrices

$$[P_0, S_0, P_1, S_1, \ldots, P_i, S_i],$$

and by adding to line 10 in Algorithm 1 the following instructions

1. $\mathcal{Z}1 = [P_0, S_0, P_1, S_1, \ldots, P_i, S_i]$ ;
2. $\mathcal{Z} = A\mathcal{Z}1 = [V_0, T_0, V_1, T_1, \ldots, V_i, T_i]$ ;
3. $\mathcal{Z}2 = \mathcal{Z}^{\dagger} R_i$;
4. $X_i^{FEnha} = X_i + \mathcal{Z}1\mathcal{Z}2$;
5. $R_i^{FEnha} = R_i - \mathcal{Z}\mathcal{Z}2$.

## 4.4 Block BiCGStab method and its enhancement

As for Gl-BiCGStab method, we will propose an improvement of the convergence of the block BiCGStab method by applying the Proposition 5.12. Two enhancements of this method are proposed, the first one will be called the block partial enhancement, denoted by PEnha-Bl-BiCGStab(k), and the second one will be called block full enhancement, denoted by FEnha-Bl-BiCGStab.

First let us recall the block version of BiCGStab (Bl-BiCGStab).

**Algorithm 8** Block Bi-Conjugate Gradient Stabilized (Bl-BiCGStab) [33]

1. $X_0 \in \mathbb{R}^{N\times m}$ guess initial matrix;
2. $R_0 = B - AX_0$, $P_0 = R_0$, $\tilde{R}_0 = R_0$;
3. for $i = 0, 1, 2 \ldots$;
4. $V_i = AP_i$;
5. $\alpha_i = \left(\tilde{R}_0^T V_i\right)^{-1} \left(\tilde{R}_0^T R_i\right)$;
6. $S_i = R_i - V_i\alpha_i$;
7. $T_i = AS_i$;
8. $\omega_i = \langle T_i, S_i\rangle_F \,/\, \langle T_i, T_i\rangle_F$;
9. $X_{i+1} = X_i + \alpha_i P_i + \omega_i S_i$;
10. $R_{i+1} = S_i - \omega_i T_i$;
11. $\beta_i = -\left(\tilde{R}_0^T V_i\right)^{-1} \left(\tilde{R}_0^T T_i\right)$;
12. $P_{i+1} = R_{i+1} + (P_i - \omega_i V_i)\beta_i$;
13. end for.

By invoking Proposition 5.12 with the residual vector $R_i$, we obtain an improvement of the Bl-BiCGStab algorithm. Thus, we will apply an orthogonal projector $\mathcal{Q}_k$ to the residual of this method. To avoid a storage problem, we use the $k$ pairs of matrices already calculated in the Bl-BiCGStab method to construct the orthogonal projector.
The partial enhancement of the convergence of the Bl-BiCGStab method (PEnha-Bl-BiCGStab(k)) is given by choosing $\mathcal{Z}_k$ equal to the k last pairs of matrices $[P_i, S_i]$, and by adding to line 10 in Algorithm 8 the following instructions

1. $\mathcal{Z}_k = [P_{i-k+1}, S_{i-k+1}, P_{i-k+2}, S_{i-k+2}, \ldots, P_i, S_i]$;
2. $\mathcal{Z}1 = A\mathcal{Z}_k$;
3. $\mathcal{Z}2 = \mathcal{Z}1^{\dagger} R_i$;
4. $X_i^{PEnha} = X_i + \mathcal{Z}_k\mathcal{Z}2$;
5. $R_i^{PEnha} = R_i - \mathcal{Z}1\mathcal{Z}2$.

Remark that $\mathcal{Z}1 = A\mathcal{Z}_k$ is given by Algorithm 8

$$\begin{aligned} \mathcal{Z}1 = A\mathcal{Z}_k &= A[P_{i-k+1}, S_{i-k+1}, P_{i-k+2}, S_{i-k+2}, \ldots, P_i, S_i] \\ &= [V_{i-k+1}, T_{i-k+1}, V_{i-k+2}, T_{i-k+2}, \ldots, V_i, T_i]. \end{aligned}$$

The full enhancement of the convergence of the Bl-BiCGStab (FEnha-Bl-BiCGStab) method is defined by choosing $\mathcal{Z}_l$ equal to the all last pairs of matrices

$$[P_0, S_0, P_1, S_1, \ldots, P_i, S_i],$$

and by adding to line 10 in Algorithm 8 the following instructions

1. $\mathcal{Z} = [P_0, S_0, P_1, S_1, \ldots, P_i, S_i]$;
2. $\mathcal{Z}1 = A\mathcal{Z}$;
3. $\mathcal{Z}2 = \mathcal{Z}1^{\dagger}R_i$;
4. $X_i^{FEnha} = X_i + \mathcal{Z}\mathcal{Z}2$;
5. $R_i^{FEnha} = R_i - \mathcal{Z}1\mathcal{Z}2$.

We notice that at each iteration we compute two matrices $P_i$ and $S_i$. The choice of the matrix $Z_i$ is crucial because we have $2km$ vectors in each matrix for the partial enhancement instead of $2k$ like in the standard case. Then, the convergence is clear in these two cases. In other words, if the number of vectors that we use to construct the orthogonal projector is large, we obtain a clearer improvement in accuracy and stability and then the convergence will be faster.

## 4.5 Numerical examples

In this section, we consider the following convection-diffusion equation

$$\begin{cases} -\Delta u - \alpha.\nabla u - \beta u = f, & \text{in } \Omega \\ u = 0, & \text{on } \partial\Omega, \end{cases}$$

where $\Omega = (0,1)^3$ and $\alpha = (\alpha_x, \alpha_y, \alpha_z)^T \in \mathbb{R}^3$. The discretization of this equation is done via centered finite differences. The obtained matrix is sparse. Then, we have used an example where the existing methods converge. We have shown numerically that the new methods give an improvement to this convergence. That is what we have shown theoretically. For all the examples we choose $\alpha = (0.5, 0.5, 0.5)^T$, $\beta = 5$ and

$$N_x = 30, N_y = 20, N_z = 20.$$

The order of the system is $N = N_x \times N_y \times N_z = 12\,000$.

In general, to compare two iterative methods in terms of convergence accuracy and stability, we need to compare the history of the norm of the residual and error vectors. Numerically, sometimes, even if the residual norm gives calculation results, the error norm does not. That is why it is necessary to check this also for the new methods too. We define the residual norm $\|r_k\|_2$ and the error norm $\|e_k\|_2$ in the standard case as follow

$$\|r_k\|_2 = \sqrt{\langle r_k, r_k \rangle} = \sqrt{r_k^T r_k}.$$

$$\|e_k\|_2 = \|x_k - xtrue\|_2 = \sqrt{(x_k - xtrue)^T (x_k - xtrue)}.$$

For the global and block case, we use the following formulas

$$\|R_k\|_F = \sqrt{\langle R_k, R_k \rangle_F} = \sqrt{\text{trace}\left(R_k^T R_k\right)}.$$

$$\begin{aligned} \|E_k\|_F &= \|X_k - Xtrue\|_F \\ &= \sqrt{\langle X_k - Xtrue, X_k - Xtrue \rangle_F} \\ &= \sqrt{\text{trace}\left((X_k - Xtrue)^T (X_k - Xtrue)\right)}, \end{aligned}$$

with *xtrue* and *Xtrue* are the exact solution in standard and block cases. We would also compare the execution time of each method, but in this case, the difference between the improved and unimproved methods is negligible (See Tables 1, 2 and 3). Because to minimize the norm of the residual, we have used the matrices and vectors already computed at each iteration.

To illustrate the efficiency of our technique we compare the BiCGStab and its enhancements methods for systems with single right-hand sides, given by Algorithm 6 with the GMRES methods. Then, we apply the classical BiCGStab and new enhanced BiCGStab(k) (partial and full enhancement of BiCGStab), denoted by PEnha-BiCGStab(k) and FEnha-BiCGStab for $k = 5$ and $k = 12$, we give the curves of residual norms and error norms. For these methods the right-hand $b$ of the system is chosen as follows

$$xtrue = \text{rand}(N, 1), \quad b = A\ xtrue,$$

where $xs$ is the solution of the considered system and the rand function creates a random N-vector for $xs$, with coefficients uniformly distributed in [0, 1] and the initial guess was taken to be zero. For this case, the tests were stopped as soon as $\| r_n \| / \| b \| \leq 10^{-10}$. Figure 1 and Figure 2 illustrate the comparison of these algorithms for residual and error norms respectively. Remark that the function randn can be also used, which creates a random matrix or vector, with real random coefficients.
For global and block methods the right hand $B$ of (5.11) is chosen as follows

$$Xtrue = rand(N, m) \quad B = A\ Xtrue,$$

the initial guess matrix equal to $zeros(N, m)$. The tests were stopped as soon as

$$\| R_n \|_F / \| B \|_F \leq 10^{-10}.$$

For the global case we compare the global BiCGStab (Gl-BiCGStab) and its enhancements, partial enhancement of global BiCGStab(k) (PEnha-Gl-BiCGStab(k)) and full enhancement of global BiCGStab (FEnha-Gl-BiCGStab) for $k = 5$, $k = 12$ and for $m = 6$, with the Gl-GMRES method. Figure 3 and Figure 4 give this comparison of residual and error norms respectively.
For the block case we compare the block BiCGStab (Bl-BiCGStab) and its enhancements, partial enhancement of block BiCGStab(s) (PEnha-Bl-BiCGStab(k)) and full enhancement of block BiCGStab (FEnha-Bl-BiCGStab) for $k = 5$, $k = 12$ and for $m = 6$, with the

Bl-GMRES method. Figure 5 and Figure 6 show this comparison of residual and error norms respectively.

### 4.5.1 BiCGStab method

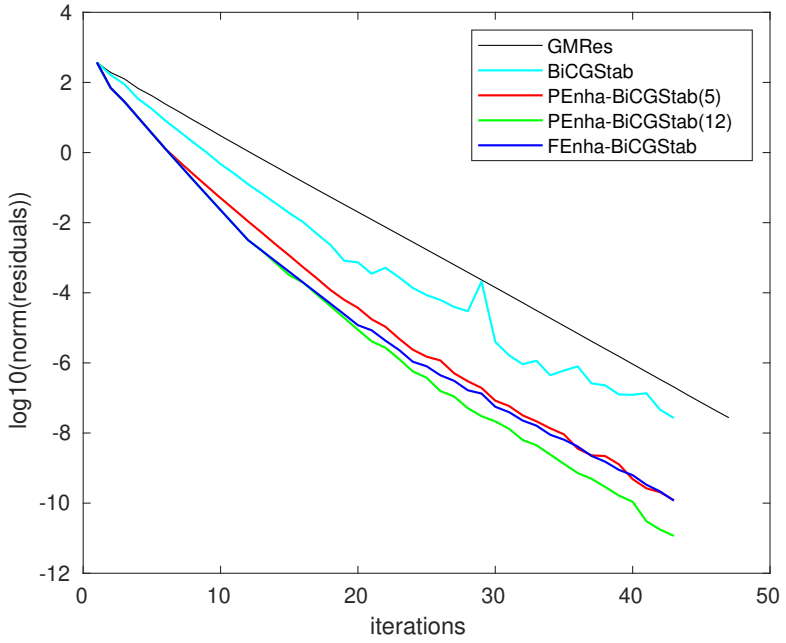


**Figure 4.1:** *The comparison between residual norms of BiCGStab, its enhancements and GMRES methods, for $k = 5$ and $k = 12$.*

| Method | CPU time (s) |
|---|---|
| BiCGStab | $5.01 \times 10^{-1}$ |
| PEnha-BiCGStab(5) | $5.48 \times 10^{-1}$ |
| PEnha-BiCGStab(12) | $6.15 \times 10^{-1}$ |
| GMRES | $5.89 \times 10^{-1}$ |

**Table 4.1:** *Comparison of CPU time in the standard case*

From the curves of Figures 4.1 and 4.2, we remark that the residual norm and the error norm of the enhanced algorithms decreases quickly comparing with the existing methods. Furthermore, for $k = 12$, the convergence is clearly faster than the case when $k = 5$.

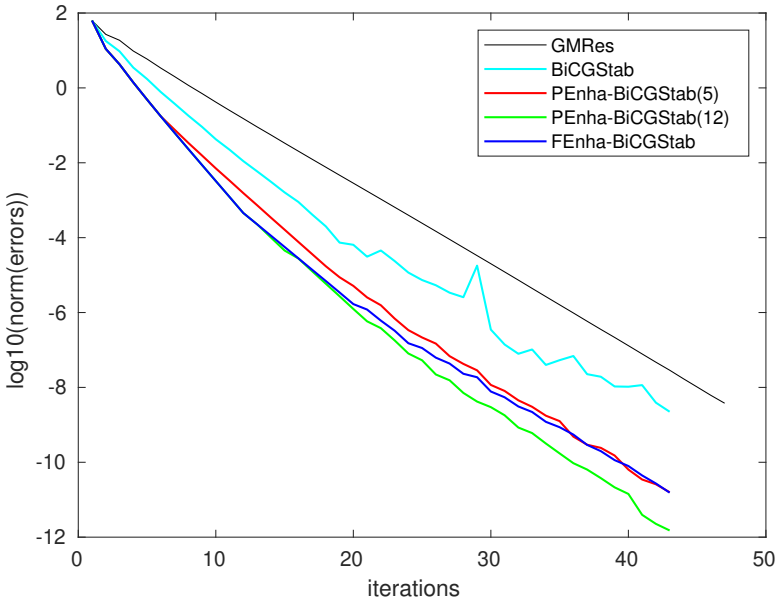


**Figure 4.2:** *The comparison between errors norms of BiCGStab, its enhancements and GMRES methods, for $k = 5$ and $k = 12$.*

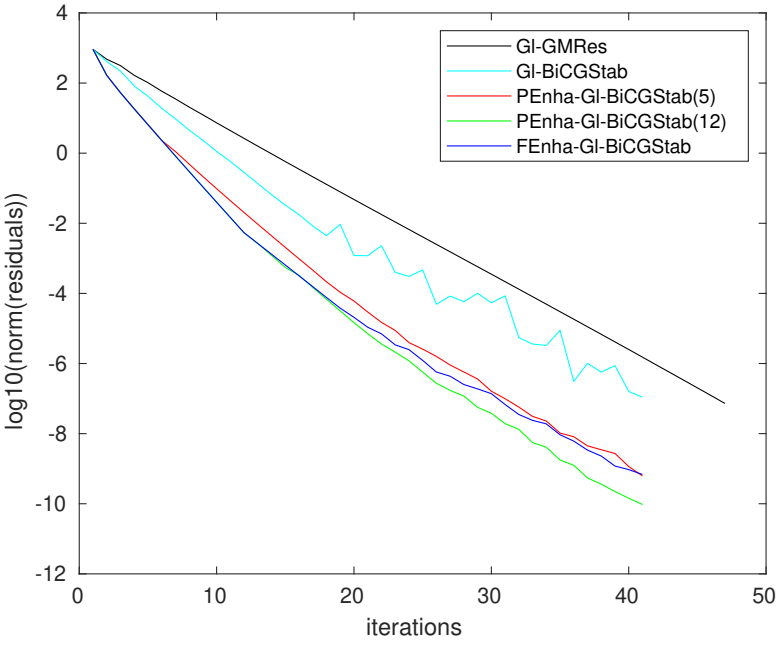


**Figure 4.3:** *Comparison between residual norms of Gl-BiCGStab, its enhancements, Gl-GMRES, for $k = 5$, $k = 12$ and for $m = 6$.*

### 4.5.2 Global BiCGStab method

In Figures 4.3 and 4.4, we observe that the enhanced solvers FEnha-Gl-BiCGStab and PEnha-Gl-BiCGStab($k$) give the best result. In this example, we can remark also that

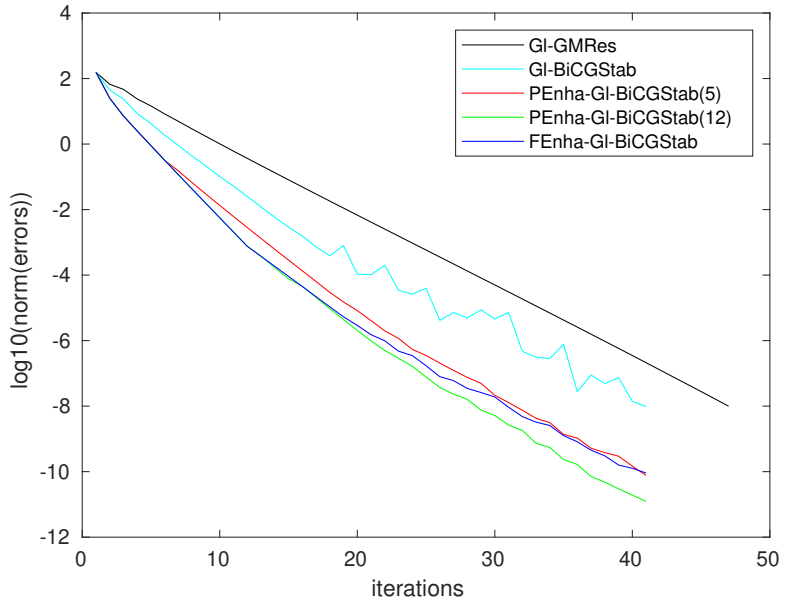


**Figure 4.4:** *Comparison between errors norms of Gl-BiCGStab, its enhancements, Gl-GMRES, for $k = 5$, $k = 12$ and for $m = 6$.*

| Method | CPU time (s) |
|---|---|
| Gl-BiCGStab | $3.69 \times 10^{0}$ |
| PEnha-Gl-BiCGStab(5) | $5.69 \times 10^{0}$ |
| PEnha-Gl-BiCGStab(12) | $7.36 \times 10^{0}$ |
| GMRES | $1.46 \times 10^{0}$ |

**Table 4.2:** *Comparison of CPU time in the global case*

for $k = 12$, the PEnha-Gl-BiCGStab($k$) is faster. A slight improvement of stability is also observed. For $k = 5$ the enhanced method is still better than Gl-BiCGStab and Gl-GMRES methods.

### 4.5.3 Block BiCGStab method

| Method | CPU time (s) |
|---|---|
| Bl-BiCGStab | $1.90 \times 10^{0}$ |
| PEnha-Bl-BiCGStab(5) | $4.01 \times 10^{0}$ |
| PEnha-Bl-BiCGStab(12) | $7.01 \times 10^{0}$ |
| GMRES | $5.80 \times 10^{-1}$ |

**Table 4.3:** *Comparison of CPU time in the block case*

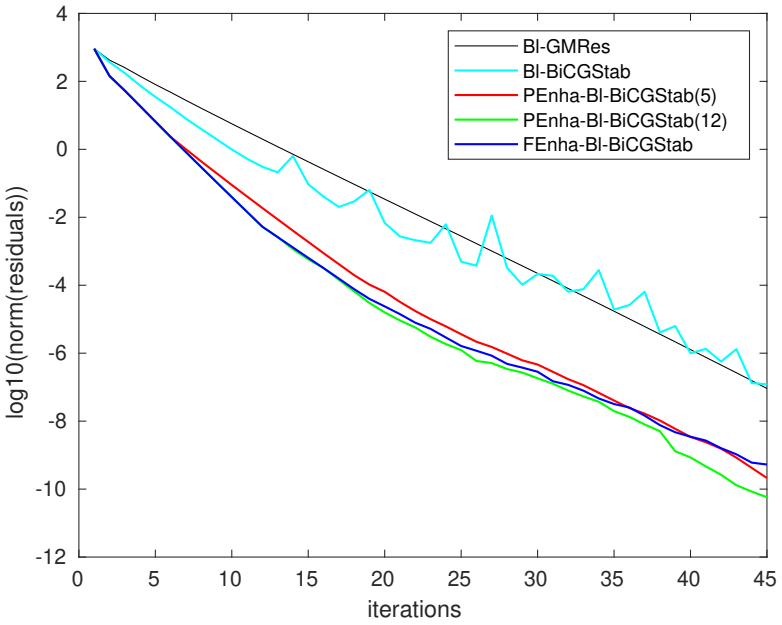


**Figure 4.5:** *Comparison between residual norms of Bl-BiCGStab, its enhancements, and Bl-GMRES, for $k = 5$, $k = 12$ and for $m = 6$.*

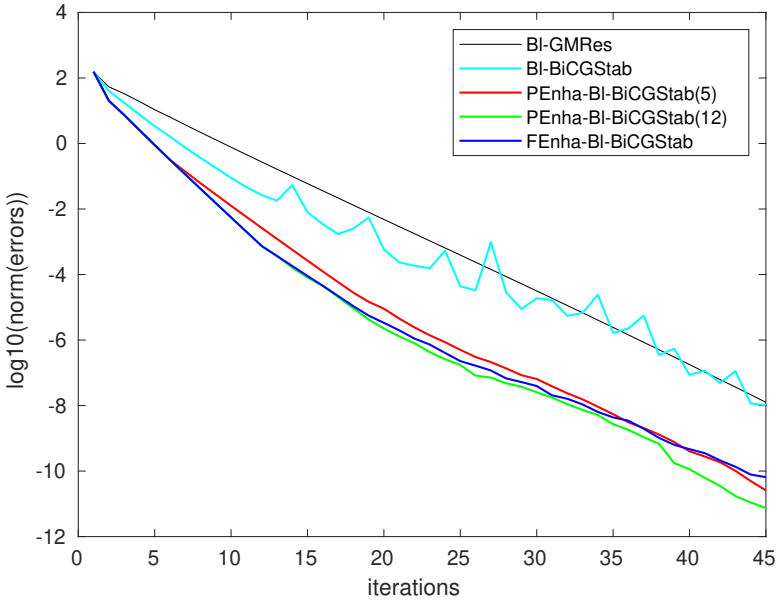


**Figure 4.6:** *Comparison between errors norms of Bl-BiCGStab, its enhancements, and Bl-GMRES, for $k = 5$, $k = 12$ and for $m = 6$.*

The curves clearly show that accuracy of the block enhanced methods is better than that of the Bl-BiCGStab and Bl-GMRES. And this is totally normal, as we have in the global and block cases, $km$ vectors to construct an orthogonal projector. The Convergence is faster as the number of vectors increases. Hence the importance of choosing matrices

$Z_i$ for constructing the orthogonal projectors in the global and block cases. The main aim of this paper is to improve the convergence of the BiCGStab method in terms of accuracy and stability with keeping the same storage properties and computation time. Tables show that when using this technique, the difference in computation time is negligible between the BiCGStab method and the improved BiCGStab method in all three cases. Here, we have also included the GMRES method as the most optimal, to show that even though we lose a little in terms of computation time, we still get a significant improvement in accuracy. So, in tables 1, 2, and 3, we focus on the comparison between the BiCGStab method and its improved version.

## 4.6 Conclusion

In this paper, we proposed a new technique to improve the convergence behavior of the BiCGStab method for the standard, global and block cases. Using orthogonal projectors, we have proposed an enhancement of the the convergence of BiCGStab method. The orthogonal projector are constructed using vectors and matrices already computed in each method to avoid storage problem and then keep the advantage storage of BiCGStab in all cases. Numerically we see that for all three cases, the enhanced algorithms of BiCGStab are more efficient and they converge faster than BiCGStab and GMRES methods with negligible deference to the turnaround time.

# Chapter 5

# An enhancement of the convergence of the IDR method


## Abstract

In this chapter, we consider a family of algorithms called IDR, based on the induced dimension reduction theorem. IDR is a family of efficient short recurrence methods, introduced by Sonneveld and Van Gijzen, for solving large systems of non-symmetric linear equations. These methods generate residual vectors that must be in a sequence of nested subspaces. We present the IDR(s) method and give two improvements of its convergence. We will also define and give the global version of the IDR(s) method, and describe the partial and complete improvement of its convergence. We will also recall the block version and give its improvements. Numerical experiments are provided to illustrate the performances of the derived algorithms compared to the well-known classical GMRES method for systems with only one right-hand side as well as the global GMRES method and the block GMRES method for systems with multiple right-hand sides.

## 5.1 Introduction

The aim of the IDR method studied in this chapter is to solve the following non-symmetric linear system

$$Ax = f, \tag{5.1}$$

where $A$ is a matrix in $\mathbb{R}^{N\times N}$ and the vectors $f$ and $x$ are in $\mathbb{R}^N$. The IDR method is a short recurrence method developed by Sonneveld and Van Gizen [91]. It is shown by Simoncini and Szyld [92], that IDR(s) is a Petrov Galerkin type method with a particular choice of the left Krylov subspace like other well-known Krylov subspace methods (FOM, GMRES, Lanczos, Hessenbeg, QMR).

The rest of this chapter is organized as follows: in the next section, we give a brief overview of the IDR(s) method. Then, we propose an improvement of the convergence of the IDR(s) algorithm using orthogonal projectors. A partial and full improvement of the IDR(s) method is proposed and will be called PEnha-IDR(s) and FEnha-IDR(s) respectively. In Section 3, we focus on the solution of linear systems with multiple right-hand sides. We will define the global version and recall the block version of the IDR(s) method, which will be called global IDR(s) (Gl-IDR(s)) and block IDR(s) (Bl-IDR(s)) methods. We will also propose two improvements of these methods, partial and full improvements, which will be called global and block IDR(s) and They noted by Gl-PEnha-IDR(s) and Bl-FEnha-IDR(s), respectively. In Section 4, we will present some numerical experiments to compare the proposed algorithms with the well-known GMRES method [101], the global GMRES method [55] and the block GMRES method [103].

Throughout this chapter, all vectors and matrices are assumed to be real and the following notation is used. First, $M^T$ represents the transpose of any matrix $M$. For two vectors $x$ and $y$ on $\mathbb{R}^N$, the inner product is $\langle x, y\rangle = x^T y$, with $\|x\| = \sqrt{\langle x, x\rangle}$ is the Euclidean norm. In the block and global cases, we consider for two matrices $X$ and $Y$ in $\mathbb{R}^{N\times m}$. The inner product is defined by $\langle X, Y\rangle_F = \mathrm{Tr}\left(X^T Y\right)$, where $\mathrm{Tr}\left(Z\right)$ denotes the trace of a square matrix $Z$. Moreover, the associated norm is the Frobenius norm noted $\|.\|_F$. We denote by $I_N$, the identity matrix of order $N$.

## 5.2 The IDR(s) method

Let $x_0$ be an initial guess and $r_0 = f - Ax_0$ is its associate residual. The Krylov subspace methods are defined by choosing at the $k$th iteration an approximation $x_k$ such that $x_k - x_0$ in $\mathcal{K}_k = \mathcal{K}_k^s(A, r_0)$. Thus, the residual

$$r_k = f - Ax_k = r_0 + A(x_k - x_0) \in \mathcal{K}_{k+1}.$$

Moreover, these methods satisfy the Petrov-Galerkin condition. Then, for a given set $\{\mathcal{L}_k\}$ of nested subspaces, the Petrov-Galerkin condition consists of imposing that the residual $r_k$ be orthogonal to the subspace $\mathcal{L}_k$, and finding an approximate solution $x_k \in x_0 + \mathcal{K}_k$ such that $r_k = f - Ax_k \perp \mathcal{L}_k$. All different choices of the left subspace $\mathcal{L}_k$ give different variant of iterative methods. For example, when $\mathcal{L}_k = \mathcal{K}_k$, we obtain the Full Orthogonal Method (FOM) (see [89]). On the other hand, when $\mathcal{L}_k = A\mathcal{K}_k$, one has a minimal residual methods, such as the well-known GMRES method.

The IDR(s) method is a variant of IDR, using $s$ shadow vectors, developed by Sonneveld and Van Gizen [91]. The subspaces used by the IDR algorithms are related to the Krylov subspace. We will first recall the definition of Krylov subspace of order $n$ associated to the matrix $A$ and the vector $r_0$ by

$$\mathcal{K}_n^s(A, r_0) = span\{r_0, Ar_0, \cdots, A^{n-1}r_0\},$$

where $r_0 = f - Ax_0$ with $x_0$ a guess initial approximation of the solution of system (5.1).

### 5.2.1 The IDR theorem

The IDR(s) method is based on the following Induction Dimension Reduction (IDR) theorem [82], which is a generalization of the original IDR theorem [91] to the complex case, we first review this theorem.

**Theorem 5.1.** *(IDR) [82]*

*Let $A$ be any matrix in $\mathbb{C}^{N\times N}$, and let $\mathcal{G}_0$ be the full Krylov space $K_N(A, r_0)$. Let $\mathcal{S}$ be any eigensubspace of $\mathbb{C}^N$ such that $\mathcal{S}$ and $\mathcal{G}_0$ do not share a nontrivial invariant subspace of $A$, and define the sequence $\mathcal{G}_j$, $j = 1, 2, \cdots$, as*

$$\mathcal{G}_j = \left(I - \omega_j A\right)\left(\mathcal{G}_{j-1} \cap \mathcal{S}\right), \tag{5.2}$$

*where the $\omega_j$'s are nonzero complex scalars. Then the following hold:*

- $\mathcal{G}_j \subset \mathcal{G}_{j-1}, \forall j > 0$.
- $\mathcal{G}_\nu = \{\mathbf{0}\}$, *for some* $\nu \leq N$.

From this theorem, we know that the dimension of the nested subspaces $\mathcal{G}_j$ decreases. If all the residual $r_n = f - Ax_n$ can be constructed in the nested subspaces $\mathcal{G}_j$, we may get the approximate solution in finite steps. At most $N + N/s$ matrix-vector products will be needed in the generic case for the IDR(s) method [91].

### 5.2.2 The IDR(s) algorithm

Consider $\mathcal{S} = \mathcal{N}\left(P^T\right)$, where $P = [p_1, p_2, \ldots, p_s]$ is a full rank matrix in $\mathbb{R}^{N\times s}$ with $s \ll N$. For all nonzero integers $j$, the IDR spaces are recursively defined as follows

$$\mathcal{G}_j = \left(I - \omega_j A\right)\left(\mathcal{G}_{j-1} \cap \mathcal{S}\right). \tag{5.3}$$

According to the IDR theorem for all $j \leq N$, we have $\mathcal{G}_j \subset \mathcal{G}_{j-1}$ and there exists $\nu \leqslant N$ such that $\mathcal{G}_\nu = \{\mathbf{0}\}$. This is why the IDR theorem can be used to develop an algorithm for solving linear systems. This is done by constructing residuals $r_n \in \mathcal{G}_j$, because according to this theorem, it is possible to generate a sequence of smallest possible subspaces. Then, the aim of Sonneveld's approach is, first, to construct subspaces $\mathcal{G}_j$, for all nonzero integer $j$ where $\omega_j$ are nonzero scalars. Then, we compute the approximate solution $x_n$ associated to the residual vector $r_n = b - Ax_n$ which is necessarily in $\mathcal{G}_j$. Thus, the residual $r_n \in \mathcal{G}_j$ can be written as follows

$$r_n = \left(I - \omega_j A\right) v_{n-1} \text{ with } v_{n-1} \in \mathcal{G}_{j-1} \cap \mathcal{S}. \tag{5.4}$$

Since, there exists $\nu$ such that $\mathcal{G}_\nu = \{\mathbf{0}\}$, it follows that there exists an integer $m$ such that $r_m \in \mathcal{G}_\nu = \{0\}$. The scalars $\omega_j$ are chosen such that $\|r_n\|$ is minimal.

Now, to compute $r_n$ at each iteration, it is necessary to compute $v_{n-1}$ and this can be done by using the fact that $v_{n-1} \in \mathcal{S}$.

In [48], Gutknecht consider

$$v_{n-1} = r_{n-1} - G_{n-1}c, \tag{5.5}$$

where for all integers $l \leq s$, $c = (\delta_l, \ldots, \delta_1)^T \in \mathbb{R}^l$ and $G_{n-1} = [g_{n-1-l}, \ldots, g_{n-2}] \in \mathbb{R}^{N\times l}$ with $g_i \in \mathcal{G}_{j-1}$, for $i = n-l-1, \cdots, n-2$. From equations (5.4) and (5.5), we get

$$r_n = r_{n-1} - \omega_j A v_{n-1} - G_{n-1}c. \tag{5.6}$$

*Remark* 5.2. In order to determine the $s$ variables $\delta_i$, the space $\mathcal{S}$ can be chosen to be the left null space of some $N \times s$ matrix $P = [p_1 \quad p_2 \quad \ldots \quad p_s]$, e. g., $\mathcal{S} = \mathcal{N}(P^T)$, which can be generated randomly, since the probability that the space $\mathcal{G}_0 \cap \mathcal{S}$ contains some eigenvectors of $A$ is zero. Then $\delta_i$ can be determined from the equation

$$P^T v_{n-1} = 0. \tag{5.7}$$

We obtain therefore the following $s \times l$ system

$$\begin{cases} p_1^T r_{n-1} - p_1^T \left( \sum_{i=1}^{l} \delta_i g_{n-i} \right) = \quad p_1^T r_{n-1} - p_1^T \left( G_{n-1} c \right) = \quad 0 \\ \qquad\qquad\qquad\qquad\qquad\qquad \vdots \\ p_s^T r_{n-1} - p_s^T \left( \sum_{i=1}^{l} \delta_i g_{n-i} \right) = \quad p_s^T r_{n-1} - p_s^T \left( G_{n-1} c \right) = \quad 0. \end{cases}$$

Under normal circumstances the previous system is uniquely solvable if $l = s$. Then, to compute all scalars $\delta_i$ for $i = 1, \ldots, s$, we need $s$ vectors in $\mathcal{G}_j$. Consequently, computing the first vector in $\mathcal{G}_j$ requires $s+1$ vectors in $\mathcal{G}_{j-1}$, and we may expect $r_n$ to be in $\mathcal{G}_j$ only for $n \geq j(s+1)$. Define the following matrices

$$G_{n-1} = \Delta R_n = \left[ \Delta r_{n-1} \quad \Delta r_{n-2} \quad \cdots \quad \Delta r_{n-s} \right], \tag{5.8}$$

and

$$\Delta X_n = \left[\Delta x_{n-1} \;\; \Delta x_{n-2} \;\; \cdots \;\; \Delta x_{n-s}\right], \tag{5.9}$$

where the forward difference operator $\Delta u_n = u_{n+1} - u_n$ is used. Then the computation of $r_n \in \mathcal{G}_j$ can be implemented by the following algorithm

$$\begin{cases} \text{Calculate } c \in \mathbb{R}^s \text{ from } \left(P^T \Delta R_n\right) c = P^T r_{n-1}, \\ v = r_{n-1} - \Delta R_{n-1} c, \\ r_n = v - \omega_j A v. \end{cases}$$

Since $\mathcal{G}_{j-1} \subset \mathcal{G}_j$, repeating these calculations will produce new residuals $r_{n+1}$, $r_{n+2}$, ... in $\mathcal{G}_j$. Once $s+1$ residuals in $\mathcal{G}_j$ have been computed, we can expect the next residual to be in $\mathcal{G}_{j+1}$. The approximation $x_n$ associated with the residual $r_n = f - Ax_n$ is given by

$$x_n = x_{n-1} + \omega_j v - \Delta X_{n-1} c.$$

Putting all the relations together, we get the IDR(s) algorithm.

V. Simoncini and D. Szyld was approved in this work [92] that IDR method can also be considered as a Petrov Galerkin type method. In this context, the left subspace considered in IDR method is the following rational Krylov subspace

$$\mathcal{W}_j = \left(\Omega_j\left(A^T\right)\right)^{-1}\mathbb{K}_j\left(A^T, Q\right),$$

where $\left(\Omega_j\left(A^T\right)\right)^{-1} = \left(\Omega_{j-1}\left(A^T\right)\right)^{-1}\left(\left(I-\omega_j A\right)^T\right)^{-1}$, $\Omega_0(A) = I$ and $\mathbb{K}_j\left(A^T, Q\right) = \sum_{i=1}^{s}\mathcal{K}_j\left(A^T, q_i\right)$. Then, this method consists of finding at the $k$th iteration, an approximation $x_k \in x_0 + \mathcal{K}_k\left(A, r_0\right)$ such that for all $k > s$ the residual $r_k$ satisfies the following orthogonality condition

$$r_k \perp \mathcal{W}_j.$$

The prototype IDR Algorithm that is described in [91] is only one of many possible IDR variants see for example [74, 6, 29, 47, 48, 81]. One of the possibilities to make alternative IDR method is different computation of the intermediate residuals. In IDR method, the residual is uniquely defined in every $s+1$ step. This step corresponds to the calculation of the first $\mathcal{G}_j$. In order to advance to $\mathcal{G}_{j+1}$, $s$ additional residuals in $\mathcal{G}_j$ should be computed. These intermediate residuals are not uniquely defined and their computation leaves freedom to derive algorithmic variants. The residuals do not depend on how the intermediate residuals are computed. The numerical stability and efficiency of the specific IDR Algorithm, however, depending on the computation of the intermediate residuals. On the other hand, to start, we have to choose the $s$ vectors $p_i$ for $i = 1, \ldots, s$ which must be linearly independent. Then, we can use any Krylov subspace method to compute these vectors or simply choose $s$ linearly independent random vectors as in [91]. While in Krylov subspace method, we start only with one vector $r_0$. For this reason, IDR method can not include in the unified Krylov subspace approach as we did in [23] for FOM, GMRES, Hessenberg, Lanczos [59], QMR, CGS [83] and Bi-CGStab [104] methods. But, for $s = 1$, we have IDR method is mathematically

**Algorithm 9** IDR(s) Algorithm [91]

1. $A \in \mathbb{R}^{N\times N}$, $x_0$, $f \in \mathbb{R}^N$, $P \in \mathbb{R}^{N\times s}$, $P = orth(P)$, $tol \in [0,1]$, $itemax > 0$, $r_0 = f - Ax_0$;
2. for $n = 0$ to $s-1$ do (build $s$ vectors in $\mathcal{G}_0$)
3. $\quad v = Ar_n$;
4. $\quad \omega = (v^T r_n)/(v^T v)$;
5. $\quad \Delta x_n = \omega r_n$; $\Delta r_n = -\omega v$;
6. $\quad r_{n+1} = r_n + \Delta r_n$; $x_{n+1} = x_n + \Delta x_n$;
7. end for
8. $\Delta R_{n+1} = [\Delta r_n, \ldots, \Delta r_0]$; $\Delta X_{n+1} = [\Delta x_n, \ldots, \Delta x_0]$;
9. $n = s$;
10. while $\|r_n\| \,/\, \|b\| > tol$ and $n < itemax$ do
11. $\quad$ for $k = 0$ to $s$ do (build $s$ vectors of $\mathcal{G}_j$)
12. $\quad\quad$ solve c from $P^T \Delta R_n c = P^T r_n$;
13. $\quad\quad$ compute $q = -\Delta R_n c$, $v = r_n - q$;
14. $\quad\quad$ if $k = 0$ then
15. $\quad\quad\quad t = Av$; $\omega = (t^T v)/(t^T t)$;
16. $\quad\quad\quad \Delta r_n = q - \omega t$; $\Delta x_n = -\Delta X_n c + \omega v$;
17. $\quad\quad$ else
18. $\quad\quad\quad \Delta x_n = -\Delta X_n c + \omega v$;
19. $\quad\quad\quad \Delta r_n = -A\Delta x_n$;
20. $\quad\quad$ end if
21. $\quad\quad r_{n+1} = r_n + \Delta r_n$; $x_{n+1} = x_n + \Delta x_n$;
22. $\quad\quad n = n + 1$;
23. $\quad\quad \Delta R_n = [\Delta r_{n-1}, \ldots, \Delta r_{n-s}]$; $\Delta X_n = [\Delta x_{n-1}, \ldots, \Delta x_{n-s}]$;
24. $\quad$ end for
25. end while.

equivalent to the BiCGStab method as shown in [48, 90] and it is related to Ml(k)Bi-CGStab method for $s > 1$ (see [93]) for more details. Then, IDR in this case can be considered as a product type method.

Now, comparing IDR with GMRES in term of accuracy, in [73], it shown that GMRES is best than IDR. But IDR is cheaper than GMRES in term of storage and time. For us, the fact to construct $s$ vectors at each iteration is not a disadvantage. In the following section, we will show how we can use these vectors to improve the convergence of the IDR method for any integer $s$.

### 5.2.3 Partial and full enhancement of the convergence of the IDR(s) method

We will propose an improvement of the convergence of the IDR(s) method. Two enhancements of this method are studied, the first one will be called partial enhancement, denoted by PEnha-IDR(s), and the second one will be called full enhancement, denoted by FEnha-IDR(s). We propose to improve the convergence of the IDR(s) method by using the following well-known result.

**Proposition 5.3.** *Consider the orthogonal projector*

$$Q_l = I - Z_l Z_l^{\dagger},$$

*where the rectangular matrix $Z_l$ is a full rank matrix in $\mathbb{R}^{N\times l}$ and $Z_l^{\dagger} = \left(Z_l^T Z_l\right)^{-1} Z_l^T$ is its pseudo-inverse (Moore-Penrose) (for more details of the pseudo inverse see [62]). Applying the projector $Q_l$ to any vector $r$ in $\mathbb{R}^N$, we obtain a new residual, which denoted by*

$$r^{Enha} = Q_l r.$$

*Then, we have*

$$\left\| r^{Enha} \right\| \leq \|r\|. \tag{5.10}$$

*Remark* 5.4. The matrix computed in Algorithm 9

$$\Delta R_n = [\Delta r_{n-1} \ \ \Delta r_{n-2} \ \ \Delta r_{n-s}],$$

is of full rank. Therefore, the pseudo-inverse of $\Delta R_n$ is well defined and its columns can be used for building the orthogonal projecteur $Q_l$.

By invoking Proposition 5.3 with the residual vector $r_n$, we obtain an improvement in the accuracy and stability of the IDR(s) algorithm. Thus, we will apply an orthogonal projector $Q_l$ to the residual of this method. However, we will fall into the storage problem. This problem can be avoided by using the $s$ vectors of $\mathcal{G}_j$ already computed in the IDR(s) method to construct the orthogonal projector.

The partial enhancement of the convergence of the IDR(s) (PEnha-IDR(s)) method is given by choosing $Z_l$ equal to the last column of $\Delta R_n$, ($l = 1$), and by adding to line 26 in Algorithm 9 the following instructions

1. $Z_1 = \Delta R_n(:, n)$;
2. $Z = Z_1^{\dagger} * r_{n+1}$;
3. $x_{n+1}^{PEnha} = x_{n+1} + \Delta X_n(:, n) * Z$;
4. $r_{n+1}^{PEnha} = r_{n+1} - \Delta R_n(:, n) * Z$;

The full enhancement of the convergence of the IDR(s) (FEnha-IDR(s)) method is defined by choosing $Z_l$ equal to $\Delta R_n$, ($l = s$), and by adding to line 26 in Algorithm 9 the following instructions

1. $Z_s = \Delta R_n$;
2. $Z = Z_s^{\dagger} * r_{n+1}$;
3. $x_{n+1}^{FEnha} = x_{n+1} + \Delta X_n * Z$;
4. $r_{n+1}^{FEnha} = r_{n+1} - \Delta R_n * Z$;

*Remark* 5.5. For building the orthogonal projector $Q_l$, we can also take some of the latest columns of the matrix $\Delta R_n$.

Using these new vectors, we obtain a new algorithm that improves the convergence of IDR algorithm which is denoted by EnhaIDR and summarized in Algorithm 9.

**Proposition 5.6.** *When s is large we obtain a remarkable enhancement of accuracy and stability.*

*Proof.* We can show this mathematically by remarking that, if we have more than $s$ linearly independent vectors, we can use them to construct an other orthogonal projector. Then, we minimize the norm of the last enhanced residual vector. This can be observed numerically in the last section. □

## 5.3 Global and block IDR(s) methods

In this section, we consider the solution of large and sparse nonsymmetric systems with multiple right-hand sides of the form

$$AX = F, \tag{5.11}$$

where the coefficient matrix $A$ is a nonsingular real matrix of order $N$, $X = [x_1\ x_2\ \ldots\ x_m]$ and $B = [f_1\ f_2\ \ldots\ f_m] \in \mathbb{R}^{N\times m}$, with $m \ll N$.

One class of solvers for solving problem (5.11) are the global methods, which are based on the use of a global projection process onto a matrix (global) Krylov subspace, including global FOM and GMRES methods [55], global BiCG and BiCGStab methods [33], global Hessenberg and CMRH methods [32].

The other class is that of the block solvers which are much more efficient when the matrix $A$ is relatively dense. The first block solvers are the block conjugate gradient (Bl-CG) and the block bi-conjugate gradient (Bl-BiCG) methods proposed in [74], for nonsymmetric problems, the Bl-BiCG, the block generalized minimal residual (Bl-GMRES) algorithm [103], the block quasi-minimum residual (Bl-QMR) algorithm [39],

**Algorithm 10** PEnhaIDR algorithm

1. $A \in \mathbb{R}^{n\times n}$, $x_0$, $f \in \mathbb{R}^n$, $Q \in \mathbb{R}^{n\times s}$, $tol \in [0,1]$, $itemax > 0$, $r_0 = f - Ax_0$;
2. For $k = 0, \ldots, s-1$, do (Construct $s$ initial vectors in $\mathcal{G}_0$)
3. $\quad u = Ar_k$; $\omega = \langle u, r_k\rangle \,/\, \langle u, u\rangle$;
4. $\quad dx_k = \omega r_k$; $dr_k = -\omega u$;
5. $\quad r_{k+1} = r_k + dr_k$; $x_{k+1} = x_k + dx_k$;
6. End for
7. $dR_{k+1} = [dr_k, \ldots, dr_0]$; $dX_{k+1} = [dx_k, \ldots, dx_0]$;
8. $k = s$
9. While $\|r_k\| > tol$ or $k < itemax$
10. $\quad$ For $j = 0, \ldots, s$, do (Construct $s$ vectors of $\mathcal{G}_j$)
11. $\quad\quad$ Solve the system $Q^T dR_k c = Q^T r_k$ and compute $u = r_k - dR_k c$;
12. $\quad\quad$ If $k = 0$, then
13. $\quad\quad\quad t = Au$;
14. $\quad\quad\quad \omega = \langle t, u\rangle \,/\, \langle t, t\rangle$;
15. $\quad\quad\quad dr_k = -dR_k c - \omega t$; $dx_k = -dX_k c + \omega u$;
16. $\quad\quad$ Else if
17. $\quad\quad\quad dx_k = -dX_k c + \omega u$; $dr_k = -Adx_k$;
18. $\quad\quad$ End if
19. $\quad\quad r_{k+1} = r_k + dr_k$; $x_{k+1} = x_k + dx_k$; $k = k+1$;
20. $\quad\quad dR_k = [dr_{k-1}, \ldots, dr_{k-s}]$; $dX_k = [dx_{k-1}, \ldots, dx_{k-s}]$;
21. $\quad\quad w_k = dR_k^{\dagger} r_{k+1}$;
22. $\quad\quad r_{k+1}^{\text{Enha}} = r_{k+1} - dR_k w_k$; $x_{k+1}^{\text{Enha}} = x_{k+1} + dX_k w_k$;
23. $\quad$ End for
24. End while

the block BiCGStab (Bl-BiCGStab) algorithm [33], the block Lanczos method [34] have been developed.

### 5.3.1 Global IDR(s) method

In this section we will recall some products that will be used to define the global version of IDR(s) method for solving the system of non-symmetric linear equations (5.11). As for the IDR(s) method, partial and full enhancement of this version will be proposed. We will recall the definition of the Kronecker product [62] and of the $\diamond$ product [14] and give some fundamental properties of the later.

A matrix system of $\mathbb{R}^{N\times m}$ is said to be F-orthonormal if it is orthonormal with respect to $\langle Y, Z\rangle_F = Tr(Y^T Z)$. For any matrix $A = (a_{i,j})$ and any matrix $B$, the Kronecker product of $A$ and $B$ is defined by $A \otimes B = [a_{i,j}B]$.

In what follows, we recall the product denoted by $\diamond$ and defined as follows [14].

**Definition 5.7.** Let $A = [A_1, A_2, \cdots, A_s]$ and $B = [B_1, B_2, \cdots, B_l]$ be matrices of dimension $N \times sm$ and $N \times lm$, respectively, where $A_i$ and $B_j$ $(i = 1, \cdots, s; j = 1, \cdots, l)$ are $N \times m$ matrices. Then the $s \times l$ matrix $A^T \diamond B$ is defined by

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

*Remark* 5.8.

1. If $m = 1$ then $A^T \diamond B = A^T B$.
2. If $m = 1$, $s = 1$ and $l = 1$, then setting $A = u \in \mathbb{R}^N$, $B = v \in \mathbb{R}^N$, we have $A^T \diamond B = u^T v \in \mathbb{R}$.
3. The matrix $A = [A_1, A_2, \cdots, A_s]$ is F-orthonormal if and only $A^T \diamond A = I_s$.
4. If $X \in \mathbb{R}^{N\times m}$, then $X^T \diamond X = \| X \|_F^2$.

We will give properties of this product combined with the Kronecker product.

**Proposition 5.9.** *Let $A, B, C \in \mathbb{R}^{N\times sm}$, $D \in \mathbb{R}^{N\times N}$, $L \in \mathbb{R}^s$ and $\alpha \in \mathbb{R}$. Then we have*

1. $(A+B)^T \diamond C = A^T \diamond C + B^T \diamond C$.

2. $A^T \diamond (B+C) = A^T \diamond B + A^T \diamond C$.

3. $(\alpha A)^T \diamond C = \alpha(A^T \diamond C)$.

4. $(A^T \diamond B)^T = B^T \diamond A$.

5. $(DA)^T \diamond B = A^T \diamond (D^T B)$.

6. $\| A^T \diamond B \|_F \leq \| A \|_F \| B \|_F$.

7. $A^T \diamond (B(L \otimes I_m)) = (A^T \diamond B)L$.

*Proof.* The first six assertions are proved in [14]. We will prove the last one. We define $A = [A_1\ A_2\ \ldots\ A_s] \in \mathbb{R}^{N \times sm}$, $B = [B_1\ B_2\ \ldots\ B_s] \in \mathbb{R}^{N \times sm}$, with $A_i$, $B_i \in \mathbb{R}^{N \times m}$ and $L = (l_1,\ l_2,\ \ldots,\ l_s)^T \in \mathbb{R}^{s \times 1}$. Then using the definition of the product $\diamond$ we get

$$\begin{aligned} A^T \diamond (B(L \otimes I_m)) &= A^T \diamond \left( \sum_{i=1}^{s} B_i l_i \right) \\ &= \sum_{i=1}^{s} \left( A^T \diamond B_i \right) l_i \\ &= \left( A^T \diamond B \right) L, \end{aligned}$$

and the result follows. □

We recall the definition of the global Krylov subspace of order $n$ associated with the matrices $A$ and $R_0$, where $R_0 = F - AX_0$ with $X_0$ an initial approximation of the solution, $X^*$, of the system (5.11).

**Definition 5.10.** The subspace $\mathcal{K}_n^g(A, R_0)$ generated by $A$ and increasing powers of $A$ applied to $R_0$

$$\mathcal{K}_n^g(A, R_0) = \left\{ \sum_{i=1}^{n} \gamma_i A^{i-1} R_0;\ \gamma_i \in \mathbb{R} \right\},$$

is called the global Krylov subspace of order $n$ associated with $A$ and $R_0$; see e.g.,[55]. It can be also defined by

$$\mathcal{K}_n^g(A, R_0) = span\left\{R_0, AR_0, \ldots, A^{n-1}R_0\right\}.$$

If we set $K_n = [R_0 \ AR_0 \ \ldots \ A^{n-1}R_0]$ and $\gamma = (\gamma_1, \gamma_2, \ldots, \gamma_s)^T$, then using the Kronecker product, the subspace $\mathcal{K}_n^g(A, R_0)$ can be written as follows

$$\mathcal{K}_k^g(A, R_0) = \left\{\sum_{i=1}^{n} A^{i-1}R_0\gamma_i;\ \gamma_i \in \mathbb{R}\right\} = \left\{K_n(\gamma \otimes I_m);\ \gamma \in \mathbb{R}^s\right\}.$$

#### 5.3.1.1 The global IDR(s) algorithm

The global version of the IDR theorem can be given as follows, whose proof is similar to that of the IDR theorem.

**Theorem 5.11.** *(global IDR)*
*Let $A$ be any matrix in $\mathbb{C}^{N\times N}$, and let $\mathcal{G}_0^g$ be the full global Krylov space $\mathcal{K}_N^g(A, R_0)$. Let $\mathcal{S}^g$ denote any eigensubspace of $\mathbb{C}^N$ such that $\mathcal{S}^g$ and $\mathcal{G}_0^g$ do not share a nontrivial invariant subspace of $A$, and define the sequence $\mathcal{G}_j^g$, $j = 1, 2, \cdots$, as follows*

$$\mathcal{G}_j^g = \left(I - \omega_j A\right)\left(\mathcal{G}_{j-1}^g \cap \mathcal{S}^g\right), \tag{5.12}$$

*where the $\omega_j$'s are nonzero complex scalars. Then the following conditions are satisfied:*

- $\mathcal{G}_j^g \subset \mathcal{G}_{j-1}^g, \forall j > 0.$
- $\mathcal{G}_\nu^g = \{\mathbf{0}\}$, *for some* $\nu \leq N$.

The global IDR(s) algorithm is an extension of the IDR(s) algorithm. It can be derived as a translation of the global IDR theorem. Assume that all column vectors of $R_{n-s}, \ldots, R_{n-1}$ belong to $\mathcal{G}_{j-1}^g$. Then, we can construct the global residual $R_n$ whose column vectors belong to $\mathcal{G}_j^g$, by defining

$$R_n = (I - \omega_j A)V_{n-1},$$

where $V_{n-1}$ is an $N \times m$ matrix such that $V_{n-1} \in \mathcal{G}_{j-1}^{g} \cap \mathcal{S}^{g}$ and the scalar parameter is obtained by minimizing the Frobenius norm of the residual $R_n$. To obtain such $V_{n-1}$, suppose that the subspace $\mathcal{S}^{g}$ can be written as follows

$$\mathcal{S}^{g} = \mathcal{N}(\mathbf{P}^T) = \{Z \in \mathbb{R}^{N\times m}; \mathbf{P}^T \diamond Z = 0_{s\times 1}\},$$

for a certain $N \times sm$ matrix $\mathbf{P}$. Let

$$V_{n-1} = R_{n-1} - \sum_{i=1}^{s} \Delta R_{n-1-i}\delta_i, \quad \text{where } \Delta R_k = R_{k+1} - R_k.$$

Then, the condition $V_{n-1} \in \mathcal{S}^{g}$ can be written

$$\mathbf{P}^T \diamond V_{n-1} = 0_{s\times 1}. \tag{5.13}$$

The coefficients $\delta_1,\ \delta_2,\ ,\ldots,\ \delta_s$ can be obtained by solving the previous equation. Define $c = (\delta_1, \delta_2, \ldots, \delta_s)^T \in \mathbb{R}^s$ and the following matrices

$$\Delta R_n^{g} = [\Delta R_{n-1}\ \ \Delta R_{n-2}\ \ \cdots\ \ \Delta R_{n-s}], \tag{5.14}$$

and

$$\Delta X_n^{g} = [\Delta X_{n-1}\ \ \Delta X_{n-2}\ \ \cdots\ \ \Delta X_{n-s}]. \tag{5.15}$$

Then, the matrix $R_n$ can be written

$$\begin{aligned} R_n &= (I - \omega_j A)V_{n-1} \\ &= (I - \omega_j A)\left(R_{n-1} - \sum_{i=1}^{s} \Delta R_{n-1-i}\delta_i\right) \\ &= R_{n-1} - \omega_j A V_{n-1} - \Delta R_n^{g}(c \otimes I_m), \end{aligned}$$

and using Proposition 5.9, equation (5.13) can be written as follows

$$\begin{aligned}
\mathbf{P}^T \diamond V_{n-1} &= \mathbf{P}^T \diamond \left( R_{n-1} - \sum_{i=1}^{s} \Delta R_{n-1-i} \delta_i \right) \\
&= \mathbf{P}^T \diamond R_{n-1} - \mathbf{P}^T \diamond \left( \Delta R_n^g (c \otimes I_m) \right) \\
&= \mathbf{P}^T \diamond R_{n-1} - \left( \mathbf{P}^T \diamond \Delta R_n^g \right) c \\
&= 0_{s \times 1}.
\end{aligned}$$

The computation of $R_n \in \mathcal{G}_j^g$ can be implemented by the following algorithm

$$\begin{cases}
\text{Calculate } c \in \mathbb{R}^s \text{ from } \left( \mathbf{P}^T \diamond \Delta R_n^g \right) c = \mathbf{P}^T \diamond R_{n-1}, \\
V_{n-1} = R_{n-1} - \Delta R_n^g (c \otimes I_m), \\
R_n = V_{n-1} - \omega_j A V_{n-1}.
\end{cases}$$

The approximation $X_n$ is obtained as follows

$$\begin{aligned}
X_n &= X_{n-1} + \omega_j V_{n-1} - \sum_{i=1}^{s} \Delta X_{n-1-i} \delta_i \\
&= X_{n-1} + \omega_j V_{n-1} - \Delta X_n^g \left( c \otimes I_m \right).
\end{aligned}$$

The scalar $\omega_j$ is given by

$$\omega_j = \frac{\langle T, V_{n-1} \rangle_F}{\langle T, T \rangle_F} = \frac{Tr(T^T V_{n-1})}{Tr(T^T T)}, \quad \text{where } T = A V_{n-1}.$$

Finally we obtain the following global IDR(s) algorithm.

#### 5.3.1.2 Partial and full enhancement of the global IDR(s) method

As for IDR(s) method, we will propose an improvement of the convergence of the global IDR(s) method. Two enhancements of these methods are studied, the first one will be called the global partial enhancement, denoted by Gl-PEnha-IDR(s), and the second one will be called global full enhancement, denoted by Gl-FEnha-IDR(s). We

**Algorithm 11** Global IDR(s) algorithm (Gl-IDR(s))

1. $A \in \mathbb{R}^{N\times N}$, $X_0$, $F \in \mathbb{R}^{N\times m}$, $\mathbf{P} \in \mathbb{R}^{N\times sm}$, $\mathbf{P} = orth(\mathbf{P})$, $tol \in [0,1]$, $itemax > 0$, $R_0 = F - AX_0$;
2. for $n = 0$ to $s-1$ do (build s matrices in $\mathcal{G}_0^g$)
3. $\quad V = AR_n$;
4. $\quad \omega = \langle V, R_n\rangle_F \,/\, \langle V, V\rangle_F$;
5. $\quad \Delta X_n = \omega R_n$; $\Delta R_n = -\omega V$;
6. $\quad R_{n+1} = R_n + \Delta R_n$; $X_{n+1} = X_n + \Delta X_n$;
7. end for
8. $\Delta R_{n+1}^g = [\Delta R_n \ \ \ldots \ \ \Delta R_0]$; $\Delta X_{n+1}^g = [\Delta X_n \ \ \ldots \ \ \Delta X_0]$;
9. $H = \mathbf{P}^T \diamond \Delta R_{n+1}^g$, $h = \mathbf{P}^T \diamond R_{n+1}$;
10. $n = s$;
11. while $\|R_n\|_F \,/\, \|B\|_F > tol$ and $n < itemax$ do
12. $\quad$ for $k = 0$ to $s$ do (build $s$ matrices of $\mathcal{G}_j^g$)
13. $\quad\quad$ solve the system $Hc = h$;
14. $\quad\quad$ compute $Q = -\Delta R_n^g\,(c \otimes I_m)$, $V = R_n - Q$;
15. $\quad\quad$ if $k = 0$ then
16. $\quad\quad\quad T = AV$ ; $\omega = \langle T, V\rangle_F \,/\, \langle T, T\rangle_F$;
17. $\quad\quad\quad \Delta R_n = Q - \omega T$; $\Delta X_n = -\Delta X_n^g\,(c \otimes I_m) + \omega V$;
18. $\quad\quad$ else
19. $\quad\quad\quad \Delta X_n = -\Delta X_n^g\,(c \otimes I_m) + \omega V$; $\Delta R_n = -A\Delta X_n$;
20. $\quad\quad$ end if
21. $\quad\quad R_{n+1} = R_n + \Delta R_n$; $X_{n+1} = X_n + \Delta X_n$;
22. $\quad\quad n = n + 1$;
23. $\quad\quad \Delta R_n^g = [\Delta R_{n-1} \ \ \ldots \ \ \Delta R_{n-s}]$; $\Delta X_n^g = [\Delta X_{n-1} \ \ \ldots \ \ \Delta X_{n-s}]$;
24. $\quad\quad \Delta h = \mathbf{P}^T \diamond \Delta R_n$;
25. $\quad\quad H(:, n) = \Delta h$;
26. $\quad$ end for
27. end while

propose to improve the convergence of the global IDR(s) method by using the following result.

**Proposition 5.12.** *Consider the orthogonal projector*

$$\mathcal{Q}_l = I - \mathcal{Z}_l \mathcal{Z}_l^{\dagger},$$

*where the rectangular matrix $\mathcal{Z}_l$ is a full rank matrix in $\mathbb{R}^{N\times lm}$ and $\mathcal{Z}_l^{\dagger} = \left(\mathcal{Z}_l^T \mathcal{Z}_l\right)^{-1} \mathcal{Z}_l^T$ its pseudo-inverse (Moore-Penrose). By applying the projector $\mathcal{Q}_l$ to any matrix $R \in \mathbb{R}^{N\times m}$, we obtain a new residual that denote by*

$$R^{Enha} = \mathcal{Q}_l R.$$

*Then, we have*

$$\left\| R^{Enha} \right\|_F \leq \|R\|_F . \tag{5.16}$$

By invoking Proposition 5.12 with the residual vector $R_n$, we obtain an improvement of the global IDR(s) algorithm. We will therefore apply an orthogonal projector $\mathcal{Q}_l$ to the residual of this method by using the $s$ matrices of $\mathcal{G}_j^g$ already computed in the global IDR(s) method to construct the orthogonal projector.

The partial improvement of the convergenceof of the global IDR(s) (Gl-PEnha-IDR(s)) method is given by choosing $\mathcal{Z}_l$ equal to the last column matrix of $\Delta R_n^g$, ($l = 1$), and by adding to line 27 in Algorithm 11 the following instructions

1. $\mathcal{Z}_1 = \Delta R_n^g(:, (n-1)m + 1 : nm)$;
2. $\mathcal{Z} = \mathcal{Z}_1^{\dagger} * R_{n+1}$;
3. $X_{n+1}^{Gl-PEnha} = X_{n+1} + \Delta X_n^g(:, (n-1)m + 1 : nm) * \mathcal{Z}$;
4. $R_{n+1}^{Gl-PEnha} = R_{n+1} - \Delta R_n^g(:, (n-1)m + 1 : nm) * \mathcal{Z}$;

The full improvement of the convergence of the global IDR(s) (Gl-FEnha-IDR(s)) method is defined by choosing $\mathcal{Z}_l$ equal to $\Delta R_n^g$, ($l = s$), and by adding to line 27 in Algorithm 11 the following instructions

1. $\mathcal{Z}_s = \Delta R_n^g$;

2. $\mathcal{Z} = \mathcal{Z}_s^\dagger * R_{n+1}$;

3. $X_{n+1}^{Gl-FEnha} = X_{n+1} + \Delta X_n^g * \mathcal{Z}$;

4. $R_{n+1}^{Gl-FEnha} = R_{n+1} - \Delta R_n^g * \mathcal{Z}$;

### 5.3.2 The block IDR(s) method

In this section we consider nonsymmetric linear systems with multiple right-hand sides (5.11). In order to propose the block version of IDR(s), we first give a variant of the IDR theorem, which is an extension of IDR theorem to the block case. We will also recall the block IDR(s) (Bl-IDR(s)), as defined in [28] and we will define the partial enhancement (PEnha-Bl-IDR(s)) method, and the full enhancement (FEnha-Bl-IDR(s)) of the convergence of this method. We first recall the block Krylov subspace of order $n$ associate to the matrices $A$ and $R_0$.

**Definition 5.13.** The subspace $\mathcal{K}_n^b(A, R_0)$ generated by $A$ and increasing powers of $A$ applied to $R_0$,

$$\mathcal{K}_n^b(A, R_0) = \left\{ \sum_{i=1}^{n} A^{i-1} R_0 \gamma_i;\ \gamma_i \in \mathbb{R}^{m\times m} \right\},$$

is called the block Krylov subspace; see e.g.,[35].

Now we will recall the definition of the block grade of $R_0$ with respect $A$ [47].

**Definition 5.14.** Let $\mathcal{B}_n(A, R_0)$ be the subspace defined as follows

$$\mathcal{B}_n(A, R_0) := \mathcal{K}_n(A, R_0(:,1)) + \ldots + \mathcal{K}_n(A, R_0(:,m)).$$

Then, the positive integer $v(A, R_0)$ defined by

$$\begin{aligned} v(A, R_0) &:= min\{n|\ \dim(\mathcal{B}_n(A, R_0))\} = \dim(\mathcal{B}_{n+1}(A, R_0)) \\ &= min\{n|\ \mathcal{B}_n(A, R_0)\} = \mathcal{B}_{n+1}(A, R_0) \end{aligned}$$

is called the block grade of $R_0$ with respect to $A$.

*Remark* 5.15. If $X^*$ is the exact block solution of $AX = B$, then

$$X^* \in X_0 + \mathcal{K}^b_{v(A,R_0)}(A, R_0).$$

#### 5.3.2.1 The block IDR(s) algorithm

Now we will recall the extension of the IDR theorem to the block case given by [28].

**Theorem 5.16.** *(Block IDR). Let $A$ be any matrix in $\mathbb{C}^{N\times N}$, and let $\mathcal{G}^b_0$ be the full block Krylov space $\mathcal{K}^b_{v(A,R_0)}(A, R_0)$. Let $\mathcal{S}^b$ denote any eigensubspace of $\mathbb{C}^N$ such that $\mathcal{S}^b$ and $\mathcal{G}^b_0$ do not share a nontrivial invariant subspace of $A$, and define the sequence $\mathcal{G}^b_j$, $j = 1, 2, \cdots$, as follows*

$$\mathcal{G}^b_j = \left(I - \omega_j A\right)\left(\mathcal{G}^b_{j-1} \cap \mathcal{S}^b\right), \tag{5.17}$$

*where the $\omega_j$'s are nonzero complex scalars. Then the following conditions are satisfied:*

- $\mathcal{G}^b_j \subset \mathcal{G}^b_{j-1}$, $\forall j > 0$.
- $\mathcal{G}^b_\nu = \{\mathbf{0}\}$, *for some* $\nu \leq v(A, R_0)$.

The block IDR(s) method is a natural extension of the IDR(s) method. It can be derived as a translation of the block IDR theorem. Suppose that all column vectors of $R_{n-s}, \ldots, R_{n-1}$ belong to $\mathcal{G}^b_{j-1}$. Then we can construct the block residual $R_n$ whose column vectors belong to $\mathcal{G}^b_j$, by defining

$$R_n = (I - \omega_j A)V_{n-1},$$

where $V_{n-1}$ is an $N \times m$ matrix whose column vectors belong to $\mathcal{G}^b_{j-1} \cap \mathcal{S}^b$ and the scalar parameter is obtained by minimizing the Frobenius norm of the block residual $R_n$. To obtain such $V_{n-1}$, suppose that the subspace $\mathcal{S}^b$ can be written as $\mathcal{S}^b = \mathcal{N}(\mathbf{P}^T)$ for some

$N \times sm$ matrix $\mathbf{P}$. Let

$$V_{n-1} = R_{n-1} - \sum_{i=1}^{s} \Delta R_{n-1-i}\gamma_i, \quad \text{where } \Delta R_k = R_{k+1} - R_k.$$

Then the condition $V_{n-1} \in \mathcal{S}^b$ can be written

$$\mathbf{P}^T V_{n-1} = 0.$$

The $m \times m$ matrices $\gamma_1, \gamma_2, \ldots, \gamma_s$ can be obtained by solving the previous equation.

The approximation is obtained as follows

$$X_n = X_{n-1} + \omega_j V_{n-1} - \sum_{i=1}^{s} \Delta X_{n-1-i}\gamma_i.$$

The scalar $\omega_j$ is given by

$$\omega_j = \frac{\langle T, V_{n-1}\rangle_F}{\langle T, T\rangle_F} = \frac{Tr(T^T V_{n-1})}{Tr(T^T T)}, \quad \text{where } T = AV_{n-1}.$$

Finally we obtain the following block IDR(s) algorithm.

#### 5.3.2.2 Partial and full enhancement of the block IDR(s) method

As for IDR(s) method, we will propose an improvement of the convergence of the block IDR(s) method by applying the proposition 5.3. Two enhancements of this method are proposed, the first one will be called the block partial enhancement, denoted by Bl-PEnha-IDR(s), and the second one will be called block full enhancement, denoted by Bl-FEnha-IDR(s).

The partial enhancement of the block IDR(s) (Bl-PEnha-IDR(s)) method is given by choosing $\mathcal{Z}_l$ equal to the last column matrix of $\Delta R_n^b$, $l = 1$, and by adding to line 27 in Algorithm 12 the following instructions

1. $\mathcal{Z}_1 = \Delta R_n^b(:, (n-1)m + 1 : nm)$;
2. $\mathcal{Z} = \mathcal{Z}_1^{\dagger} * R_{n+1}$;
3. $X_{n+1}^{Bl-PEnha} = X_{n+1} + \Delta X_n^b(:, (n-1)m + 1 : nm) * \mathcal{Z}$;
4. $R_{n+1}^{Bl-PEnha} = R_{n+1} - \Delta R_n^b(:, (n-1)m + 1 : nm) * \mathcal{Z}$;

The full enhancement of the convergence of the block IDR(s) (Bl-FEnha-IDR(s)) method is defined by choosing $\mathcal{Z}_l$ equal to $\Delta R_n^b$, $l = s$, and by adding to line 27 in Algorithm 12 the following instructions

1. $\mathcal{Z}_s = \Delta R_n^b$;
2. $\mathcal{Z} = \mathcal{Z}_s^{\dagger} * R_{n+1}$;
3. $X_{n+1}^{Bl-FEnha} = X_{n+1} + \Delta X_n^b * \mathcal{Z}$;
4. $R_{n+1}^{Bl-FEnha} = R_{n+1} - \Delta R_n^b * \mathcal{Z}$;

Using this notations and formulas, we propose a new algorithm named for us Bl-EnhaIDR, which improve the convergence of the Bl-IDR algorithm

*Remark* 5.17. We will see in the numerical examples that, the enhancement is clearly seen in the block global case. In fact, we assume that any breakdown occur, we use the $s$ blocks matrices $\Delta R\,(:, (j-1)p + 1 : jm)$ of size $n \times m$ which form a family of independent blocks considering the inner product $\langle ., . \rangle_F$.Then, as we said in section 3, if

**Algorithm 12** Block IDR(s) algorithm (Bl-IDRs))

1. $A \in \mathbb{R}^{N\times N}$, $X_0$, $F \in \mathbb{R}^{N\times m}$, $\mathbf{P} \in \mathbb{R}^{N\times sm}$, $\mathbf{P} = orth(\mathbf{P})$, $tol \in [0,1]$, $itemax > 0$, $R_0 = F - AX_0$;
2. for $n = 0$ to $s-1$ do (build s matrices of $\mathcal{G}_0^b$)
3. $\quad V = AR_n$;
4. $\quad \omega = \langle V, R_n\rangle_F \,/\, \langle V, V\rangle_F$;
5. $\quad \Delta X_n = \omega R_n$; $\Delta R_n = -\omega V$; $R_{n+1} = R_n + \Delta R_n$; $X_{n+1} = X_n + \Delta X_n$;
6. end for
7. $\Delta R_{n+1}^b = [\Delta R_n \ \ \ldots \ \ \Delta R_0]$; $\Delta X_{n+1}^b = [\Delta X_n \ \ \ldots \ \ \Delta X_0]$;
8. $H = \mathbf{P}^T \Delta R_{n+1}^b$, $h = \mathbf{P}^T R_{n+1}$;
9. $n = s$;
10. while $\|R_n\|_F \,/\, \|B\|_F > tol$ and $n < itemax$ do
11. $\quad$ for $k = 0$ to $s$ do (build $s$ matrices of $\mathcal{G}_j^b$)
12. $\quad\quad$ solve the system $HC = h$;
13. $\quad\quad$ compute $Q = -\Delta R_n^b C$, $V = R_n - Q$;
14. $\quad\quad$ if $k = 0$ then
15. $\quad\quad\quad T = AV$ ; $\omega = \langle T, V\rangle_F \,/\, \langle T, T\rangle_F$;
16. $\quad\quad\quad \Delta R_n = Q - \omega T$; $\Delta X_n = -\Delta X_n^b C + \omega V$;
17. $\quad\quad$ else
18. $\quad\quad\quad \Delta X_n = -\Delta X_n^b C + \omega V$; $\Delta R_n = -A\Delta X_n$;
19. $\quad\quad$ end if
20. $\quad\quad R_{n+1} = R_n + \Delta R_n$; $X_{n+1} = X_n + \Delta X_n$;
21. $\quad\quad n = n + 1$;
22. $\quad\quad \Delta R_n^b = [\Delta R_{n-1} \ \ \ldots \ \ \Delta R_{n-s}]$; $\Delta X_n^b = [\Delta X_{n-1} \ \ \ldots \ \ \Delta X_{n-s}]$;
23. $\quad\quad \Delta h = \mathbf{P}^T R_n$;
24. $\quad\quad H(:, (n-1)m + 1 : nm) = \Delta h$;
25. $\quad$ end for
26. end while

**Algorithm 13** Block PEnhaIDR method

1. $R_0 = F - AX_0$; $\mathbf{Q} \in \mathbb{R}^{n \times sp}$;
2. For $i = 0$ to $s - 1$ do
3. $\quad V = AR_i$, $\omega = \frac{\langle V, R_i \rangle_F}{\langle V, V \rangle_F}$;
4. $\quad \Delta X_i = \omega R_i$; $\Delta R_i = -\omega V_i$;
5. $\quad X_{i+1} = X_i + \Delta X(:, ip + 1 : (i+1)m)$; $R_{i+1} = R_i + \Delta R(:, ip + 1 : (i+1)m)$;
6. End for
7. $j = 1$; $i = s$;
8. $M = \mathbf{Q}^H \Delta R$; $h = \mathbf{Q}^H R_i$;
9. While $\max_{j=1:m} \frac{\|R_i(:,j)\|}{\|B_0(:,j)\|} > \epsilon$ do
10. $\quad$ For $k = 0$ to $s$ do
11. $\quad\quad$ Solve $c$ from $Mc = h$;
12. $\quad\quad H = -\Delta R c$;
13. $\quad\quad V = R_i + H$;
14. $\quad\quad$ If $k = 0$ then
15. $\quad\quad\quad T = AV_i$; $\omega = \frac{\langle T, V \rangle_F}{\langle T, V \rangle_F}$;
16. $\quad\quad\quad \Delta R_i = H - \omega T$; $\Delta X_i = -\Delta X c + \omega V$;
17. $\quad\quad$ Else
18. $\quad\quad\quad \Delta X_i = -\Delta X c + \omega V$; $\Delta R_i = -A \Delta X_i$;
19. $\quad\quad$ End if
20. $\quad\quad R_{i+1} = R_i + \Delta R_i$; $X_{i+1} = X_i + \Delta X_i$;
21. $\quad\quad D_{i+1} = \Delta R_i^{\dagger} R_{i+1}$
22. $\quad\quad R_{i+1}^{\text{Enha}} = R_{i+1} - \Delta R_i D_{i+1}$; $X_{i+1}^{\text{Enha}} = X_{i+1} + \Delta X_i D_{i+1}$
23. $\quad\quad \Delta p = P^H \Delta R_j$;
24. $\quad\quad M_j = \Delta p$;
25. $\quad\quad h = h + \Delta p$;
26. $\quad\quad i = i + 1$; $j = j + 1$;
27. $\quad\quad j = (j - 1) \bmod s + 1$ with 'mod' is the modulo operation
28. $\quad$ End for
29. End while

the number of vectors that we use to construct the orthogonal projector is large, we obtain a clearer improvement of accuracy and stability. This is the case because we have $[\Delta R_{k-1}, \cdots, \Delta R_{k-s}]$ of size $n \times sm$.

Even if their approaches are different, IDR and GMRES methods solve nonsymmetric linear systems. The difference between these methods is in cost, storage and accuracy. GMRES method is a long-recurrence method this is why it suffers from storage problems while IDR method is a short-recurrence method. But GMRES method remains the best in terms of accuracy because it is an orthogonal projection method (see [81]). For this reason, we tried to give the orthogonality property for IDR method to achieve the accuracy and stability of GMRES method and keep the fact that IDR method is better than GMRES method in term of time and storage. Therefore, to compare EnhaIDR with GMRES in term of memory requirements and time, it is enough to compare IDR with GMRES because the difference between IDR method and its enhancement is only the calculation of the new residual

$$r_{k+1}^{\text{Enha}} = r_{k+1} - dR_k dR_k^{\dagger} r_{k+1}.$$

## 5.4 Numerical experiments

In this section, we consider the following convection-diffusion equation

$$\begin{cases} -\Delta u - \alpha.\nabla u - \beta u = f, & \text{in } \Omega \\ u = 0, & \text{on } \partial\Omega, \end{cases}$$

where $\Omega = (0,1)^3$ and $\alpha = (\alpha_x, \alpha_y, \alpha_z)^T \in \mathbb{R}^3$. The discretization of this equation is done via centered finite differences with the standard 7-point stencil in three dimensions. For all the examples we choose $\alpha = (0.5, 0.5, 0.5)^T$, $\beta = 5$ and

$$N_x = 30, N_y = 20, N_z = 20.$$

The order of the system is then, $N = N_x \times N_y \times N_z = 12\,000$.

To illustrate the efficiency of our technique we compare the enhanced IDR(s) methods for systems with one right-hand side, given by Algorithm 9 with the GMRES method. Then, we apply the classical IDR(s) and new enhanced IDR(s) (partial and full enhancement IDR(s)), denoted by PEnha-IDR(s) and FEnha-IDR(s) for different values of $s$. For these methods the shadow vectors $P$ and the right-hand $b$ of (5.1) are chosen as follows

$$P = \text{orth}(\text{rand}(N, s)), \quad b = rand(N, 1),$$

where the rand function creates an $N \times s$ random matrix for $P$ and a random N-vector for $b$, with coefficients uniformly distributed in [0, 1] and the initial guess was taken to be zero. For this case, the tests were stopped as soon as $\| r_n \| / \| b \| \leq 10^{-10}$. Fig 4.1 and Fig 4.2 illustrate the comparison of these algorithms.
For global and block methods the shadow matrix $\mathbf{P}$ and the right hand $B$ of (5.11) are chosen as follows

$$\mathbf{P} = \text{orth}(\text{rand}(N, sm)), \quad B = rand(N, m),$$

the initial guess matrix equal to $zeros(N, m)$. The tests were stopped as soon as

$$\| R_n \|_F / \| B \|_F \leq 10^{-10}.$$

For the global case we compare the global IDR(s) (Gl-IDR(s)) and its enhancements, global partial enhancement IDR(s) (Gl-PEnha-IDR(s)) and global full enhancement IDR(s) (Gl-FEnha-IDR(s)) for different values of $s$ and $m$. Fig 4.3-Fig 4.6 give this comparison. For the block case we compare the block IDR(s) (Bl-IDR(s)) and its enhancements, block partial enhancement IDR(s) (Bl-PEnha-IDR(s)) and block full enhancement IDR(s) (Bl-FEnha-IDR(s)) for different values of $s$ and $m$. Fig 4.7-Fig 4.10 show this comparison.

### 5.4.1 IDR(s) method

We consider different values of $s$. The following figures illustrate the cases where $s = 3$ and $s = 7$ and where $s = 8$ and $s = 12$.

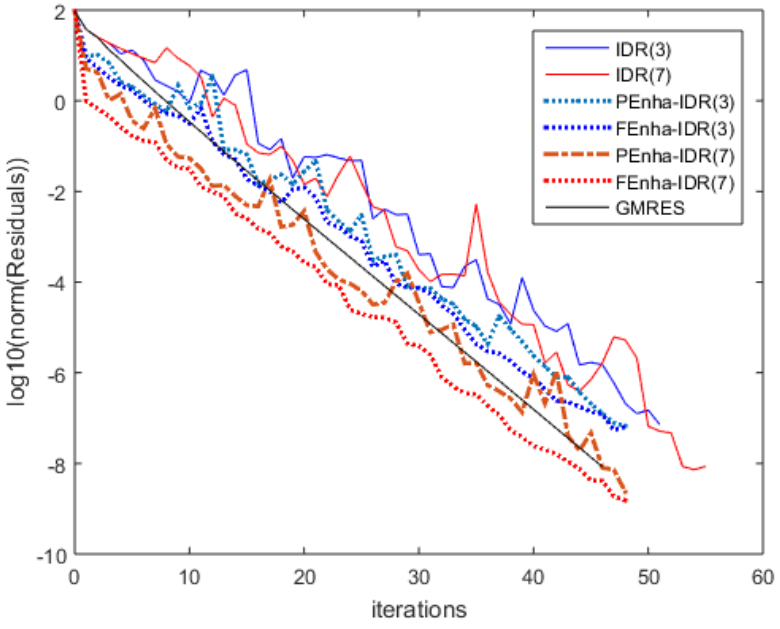


**Figure 5.1:** *The comparison between residual norms of IDR(3), IDR(7), PEnha-IDR(3), FEnha-IDR(3), PEnha-IDR(7), FEnha-IDR(7) and GMRES methods.*

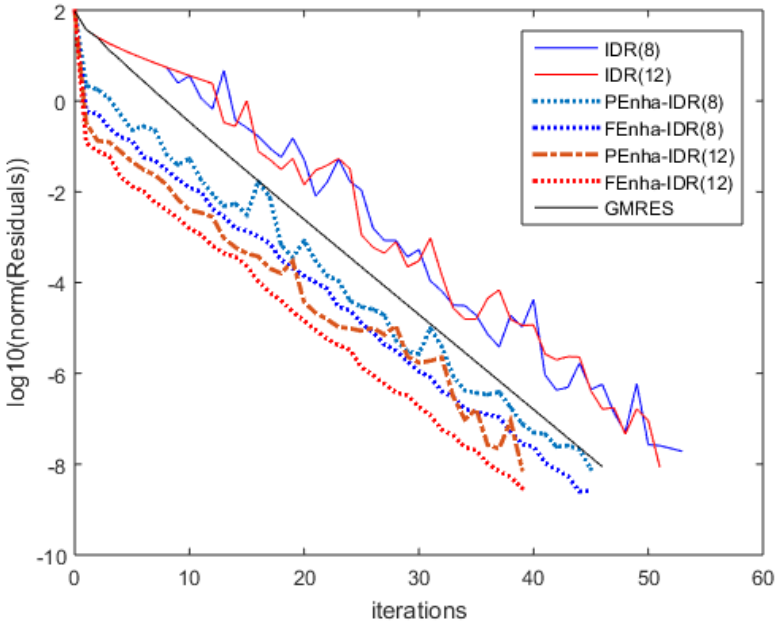


**Figure 5.2:** *The comparison between residual norms of IDR(8), IDR(12), PEnha-IDR(8), FEnha-IDR(8), PEnha-IDR(12), FEnha-IDR(12) and GMRES methods.*

**Table 6.2: Numerical results using some MatrixMarket examples.**

| MatrixMarket | | IDR | IDREnha | GMRES |
|---|---|---|---|---|
| pde225 | RES | 7.64e-11 | 1.12e-12 | 6.92e-11 |
| itemax=100 | CPU | 4.41e-03 | 1.82e-02 | 3.41e-02 |
| gr3030 | RES | 7.79e-11 | 1.99e-12 | 7.36e-11 |
| itemax=60 | CPU | 1.43e-02 | 3.55e-02 | 5.24e-02 |
| jpwh991 | RES | 2.13e-08 | 2.29e-10 | 1.84e-09 |
| itemax=70 | CPU | 1.51e-02 | 3.1e-02 | 6.25e02 |
| pde2961 | RES | 1.81e-11 | 7.11e-12 | 9.61e-11 |
| itemax=260 | CPU | 1.51e-2 | 1.83e-01 | 6.21e-2 |
| jagmesh1 | RES | 4.10e-02 | 4.55e-03 | 6.58e-04 |
| itemax=300 | CPU | 5.51e-02 | 1.62e-01 | 4.41e-01 |
| sherman1 | RES | 9.81e-11 | 9.91e-12 | 9.45e-11 |
| itemax=400 | CPU | 4.22e-02 | 1.81e-01 | 1.15e02 |
| nos3 | RES | 1.38e-09 | 9.11e-11 | 1.21e-10 |
| itemax=400 | CPU | 4.01e-02 | 1.21e-02 | 9.67e-01 |
| cavity05 | RES | 9.17e-11 | 1.95e-11 | 8.87e-11 |
| itemax=500 | CPU | 1.26e-01 | 2.33e-01 | 4.89e-01 |
| cavity10 | RES | 4.71e-09 | 1.33e-11 | 9.74e-11 |
| itemax=600 | CPU | 4.11e-01 | 7.56e-01 | 1.54e00 |
| watt1 | RES | 8.15e-10 | 5.46e-11 | 9.91e-11 |
| itemax=300 | CPU | 3.74e-01 | 3.93e-01 | 6.12e-01 |
| add32 | RES | 3.45e-11 | 1.75e-12 | 8.82e-11 |
| itemax=100 | CPU | 6.32e-2 | 3.98e-1 | 3.52e-01 |
| rdb2048 | RES | 2.98e-09 | 8.12e-11 | 9.58e-11 |
| itemax=100 | CPU | 1.24e-01 | 2.94e-01 | 1.15e01 |
| cdde1 | RES | 2.32e-11 | 7.94e-13 | 9.87e-11 |
| itemax=100 | CPU | 2.50e-2 | 6.10e-02 | 2.11e-01 |
| orsreg1 | RES | 5.40e-09 | 5.74e-10 | 8.32e-10 |
| itemax=400 | CPU | 2.31e-01 | 5.20e-01 | 3.45e01 |

In Table 5.4.1, we present different results using GMRES, IDR and IDREnha methods applying to all matrix market problems used by Meurant in his recently book [73]. In all this examples, we choose $s = 6$, the shadow vectors are chosen as follow

$$Q = \text{orth}(\text{rand}(n, s)).$$

The initial guess equal to zero and the stopping criterion was $\|r_k\| \leq 10^{-8}$. Then, If we compare the CPU time and the last residual norm of each method, we deduce that, this result confirm the fact that the derived algorithm gives best accuracy comparing with GMRES method and keep the fact that IDR method is better in term of time and storage.

### 5.4.2 Global IDR(s) method

We consider different values of $s$ and of $m$. We compare the different global IDR(s) methods and their enhancements with Gl-GMRES method.

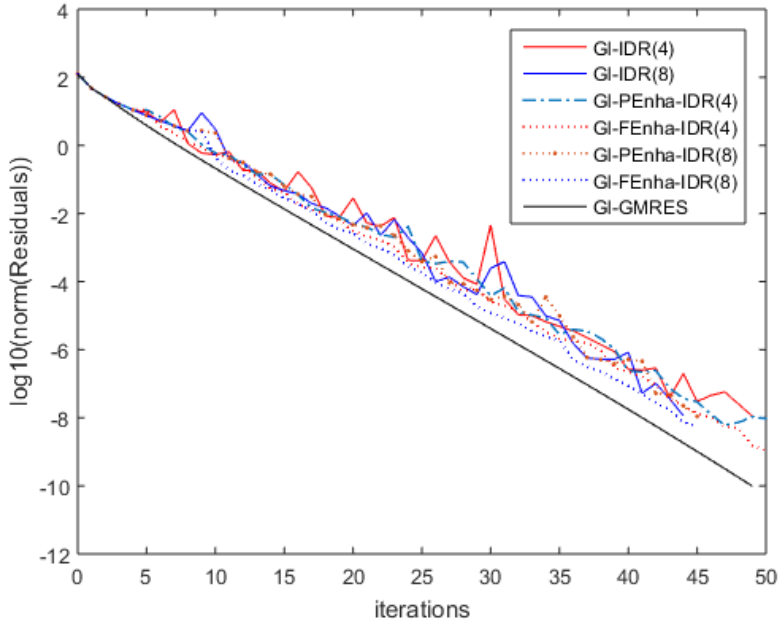


**Figure 5.3:** *The comparison between residual norms of Gl-IDR(4), Gl-IDR(8), Gl-PEnha-IDR(4), Gl-FEnha-IDR(4), Gl-PEnha-IDR(8), Gl-FEnha-IDR(8) and Gl-GMRES for $m = 4$.*

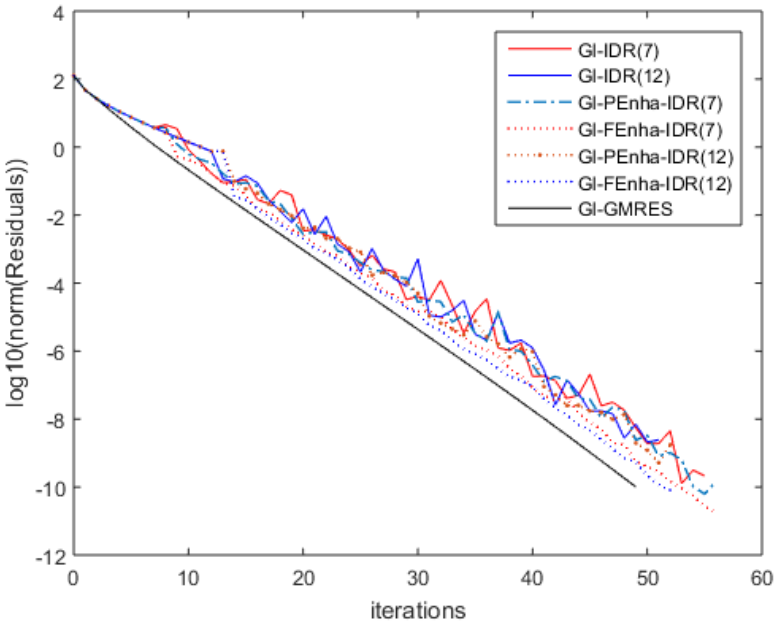


**Figure 5.4:** *The comparison between residual norms of Gl-IDR(7), Gl-IDR(12), Gl-PEnha-IDR(7), Gl-FEnha-IDR(7), Gl-PEnha-IDR(12), Gl-FEnha-IDR(12) and Gl-GMRES for* $m = 4$.

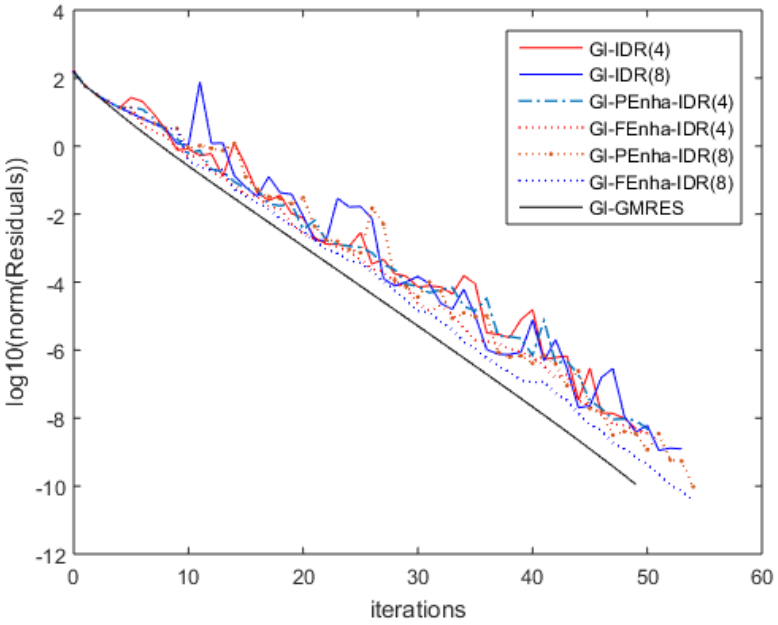


**Figure 5.5:** *The comparison between residual norms of Gl-IDR(4), Gl-IDR(8) Gl-PEnha-IDR(4), Gl-FEnha-IDR(4), Gl-PEnha-IDR(8), Gl-FEnha-IDR(8) and Gl-GMRES for* $m = 6$.

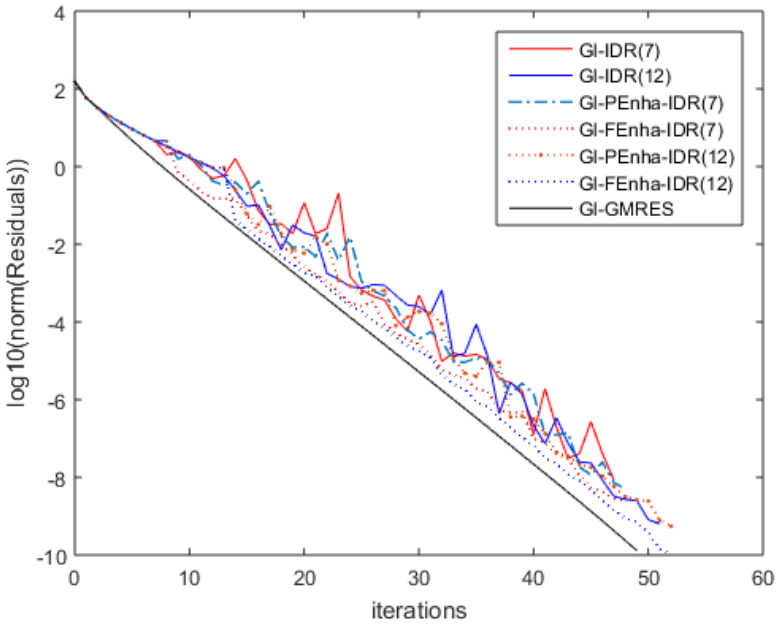


**Figure 5.6:** *The comparison between residual norms of Gl-IDR(7), Gl-IDR(12), Gl-PEnha-IDR(7), Gl-FEnha-IDR(7), Gl-PEnha-IDR(12), Gl-FEnha-IDR(12) and Gl-GMRES for* $m = 6$.

### 5.4.3 Block IDR(s) method

We consider different values of $s$ and of $m$. We compare Bl-IDR(s), Bl-PEnha-IDR(s), Bl-FEnha-IDR(s) with Bl-GMRES.

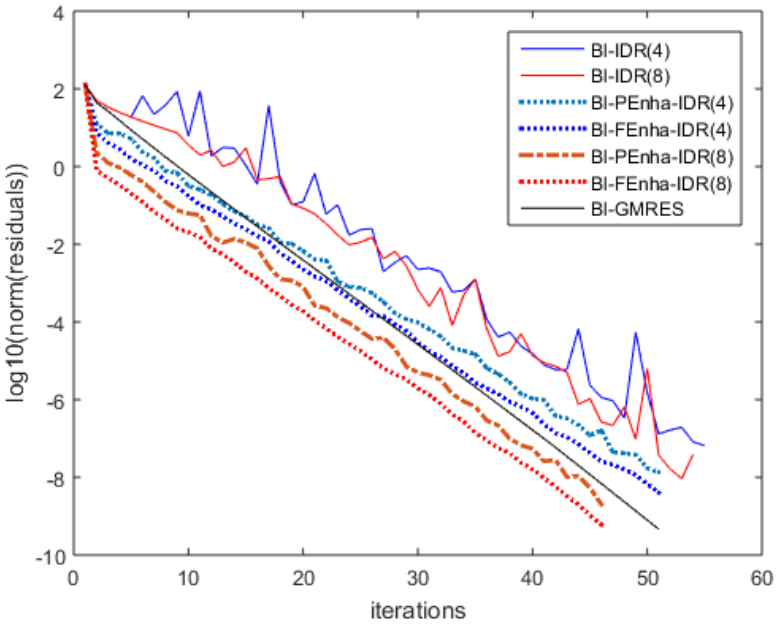


**Figure 5.7:** *The comparison between residual norms of Bl-IDR(4), Bl-IDR(8), Bl-PEnha-IDR(4), Bl-FEnha-IDR(4), Bl-PEnha-IDR(8), Bl-FEnha-IDR(8) and Bl-GMRES for* $m = 4$.

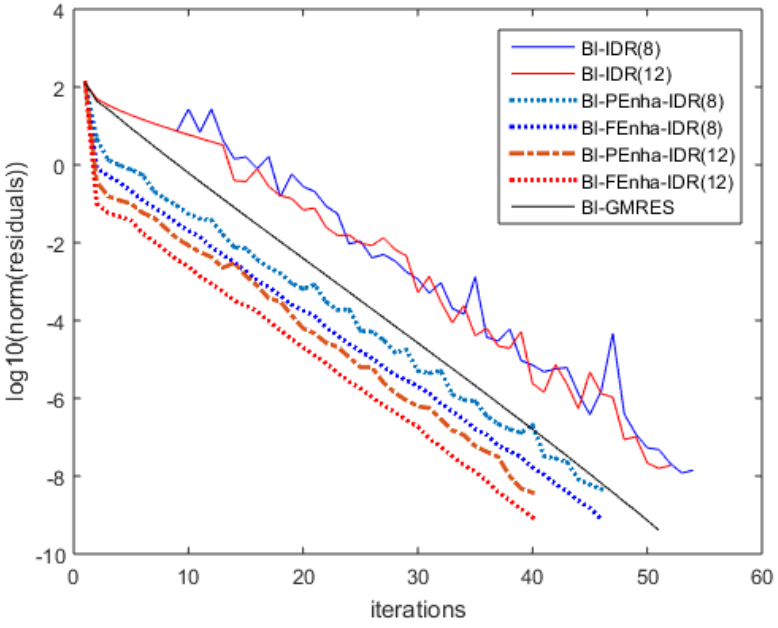


**Figure 5.8:** *The comparison between residual norms of Gl-IDR(8), Gl-IDR(12), Bl-PEnha-IDR(8), Bl-FEnha-IDR(8), Bl-PEnha-IDR(12), Bl-FEnha-IDR(12) and Bl-GMRES for* $m = 4$.

In all these figures, we remark that the derived methods in standard, global and block cases give more precision than IDR. If we compare also the smoothness of all curves,

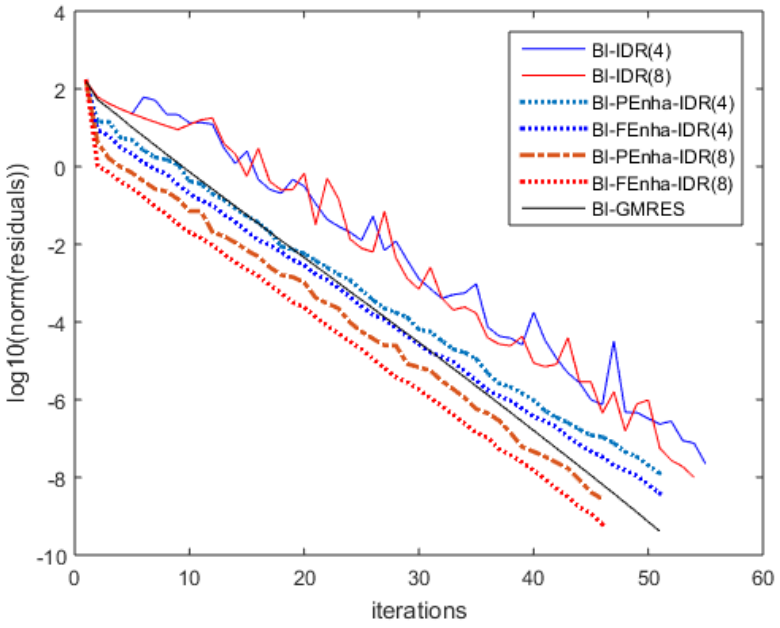


**Figure 5.9:** *The comparison between residual norms of Bl-IDR(4), Bl-IDR(8), Bl-PEnha-IDR(4), Bl-FEnha-IDR(4), Bl-PEnha-IDR(8), Bl-FEnha-IDR(8) and Bl-GMRES for* $m = 6$.

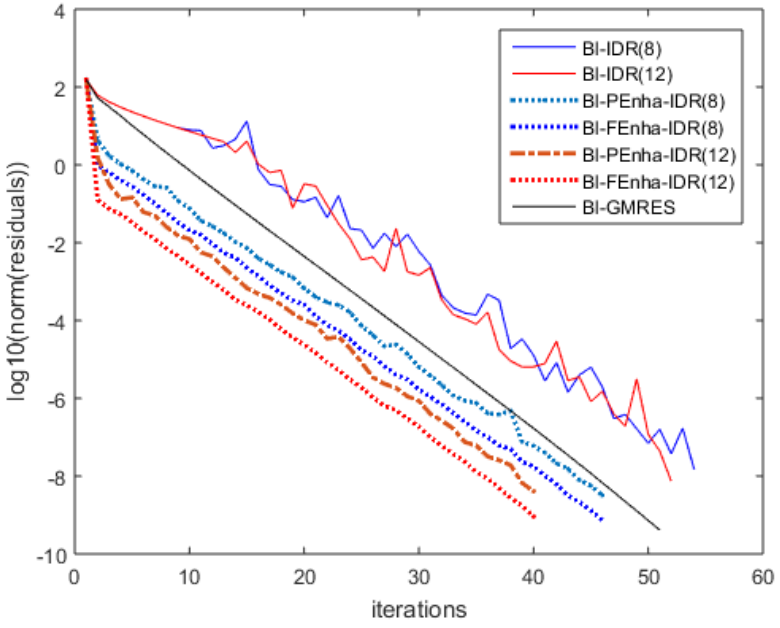


**Figure 5.10:** *The comparison between residual norms of Bl-IDR(8), Bl-IDR(12), Bl-PEnha-IDR(8), Bl-FEnha-IDR(8), Bl-PEnha-IDR(12), Bl-FEnha-IDR(12) and Bl-GMRES for* $m = 6$.

we remark that the enhanced methods are more stable than IDR method. For $s > 8$, the Enhanced IDR gives a better precision comparing with IDR. In this case, the enhanced

method can also achieves GMRES method in term of accuracy and stability and we keep the fact that our algorithm is better in term of time and storage. . Then, we conclude from this example that, we have a clearer enhancement of accuracy when $s$ increase.

## 5.5 Conclusion

In this chapter, we proposed a new technique to improve the convergence behavior of the IDR(s) method for the standard, global and block cases. Using the $s$ linearly independent vectors already computed, we have constructed orthogonal projectors to improve the convergence of the IDR(s) method. Furthermore, we have shown numerically that these methods are as efficient as the GMRES method for the standard and block cases. For the global case, we have given the global version of the IDR(s) and its improvement. The derived algorithms are also as efficient as the global GMRES method in term of the precision.

# Achievements and conclusion

In this thesis we have presented a comprehensive framework for studying Krylov subspace methods, explored their mathematical properties and convergence behaviour, and discussed techniques to enhance their performance. The paper covers various aspects including the minimal polynomial of matrix $A$, the relationship between different methods, the role of generalized inverses, and the use of product methods. The provided numerical examples further support the analysis and conclusions of the research. We introduce a comprehensive framework for studying Krylov subspace methods used to solve linear systems of the form $Ax = f$, where $A$ is a matrix, $x$ is the unknown vector, and $f$ is the right-hand side vector. The objective of these methods is to achieve convergence within a specified number of iterations, denoted as $m$.

The minimal polynomial $\Phi_m$ of matrix $A$, associated with the initial residual $r_0 = f - Ax_0$, is a key focus of analysis in the paper. The degree of $\Phi_m$ is $m$, and the properties of this minimal polynomial play a crucial role in the convergence behaviour of the Krylov subspace methods. We establish that Petrov-Galerkin methods and minimal seminorm methods are specific cases of the broader framework of Krylov subspace methods. Additionally, it is demonstrated that minimal seminorm methods satisfy implicit Petrov-Galerkin conditions. In this thesis, we present a general formulation for the iterates of Krylov subspace methods based on generalized inverses. The choice of a specific left inverse and the construction method of the Krylov basis are important factors that differentiate various Krylov subspace methods. The mathematical properties of these methods are described and analysed, with emphasis on their dependency on two matrices. The thesis proves that specific instances of Krylov subspace methods, such as CMRH (Conjugate Minimum Residual with Hessenberg matrix) and QMR

(Quasi-Minimal Residual), satisfy implicit Petrov-Galerkin orthogonality conditions. Techniques for improving the convergence behavior of Krylov subspace methods by carefully selecting vectors in their implementations are explored. The aim is to deepen the understanding of these methods, provide insights into their convergence properties, and identify potential enhancements. We also discuss Krylov methods that are product methods, where the $k$th residual $r_k$ associated with the approximation $x_k$ of the exact solution is expressed as $r_k = \Psi_k(A)\Phi_k(A)r_0$. Here, $\Psi_k$ is a polynomial of fixed or variable degree. Specific choices of $\Psi_k$, including local convergence, smoothing, fixed memory, and cost considerations for each iteration, are examined. Enhancements of product methods such like CGS BiCGStab and IDR(s) method are presented in the thesis and. In conclusion, we present a comprehensive framework for studying Krylov subspace methods, investigates their mathematical properties and convergence behaviour, explores techniques for improvement, and provides numerical examples to demonstrate the effectiveness of the proposed algorithms.

# Conclusion

Pour conclure, dans cette thèse, on a étudié les versions standard et par blocs de toutes les méthodes de sous espace de Krylov pour la résolution d'un système linéaire $AX = F$ avec $A$ une matrice supposée inversible dans $\mathbb{R}^{n\times n}$, $F$ et $X$ deux matrices de $\mathbb{R}^{n\times s}$ avec $s \ll n$. Pour $s = 1$, on a proposé une approche unifiée pour les méthodes standard du types Krylov. Cette approche est basée sur le fait que toutes les méthodes de Krylov calculent les coefficients de polynôme minimal de la matrice $A$ pour un vecteur résidu initial. En effet, le vecteur résidu de toute méthode de Krylov s'écrit comme un polynôme $P_k$ appliquée à la matrice $A$ pour un vecteur initial $r_0$. Si la méthode considérée converge après $m$ itérations le polynôme peut se décomposer en produit de deux autres polynômes $P_m = Q_m M_m$ avec $M_m$ est le polynôme minimal de $A$ pour le vecteur résidu initial. En utilisant le calcul récursif de l'inverse à gauche de la matrice de Krylov, on a pu retrouver la plupart des méthodes de Krylov et développer un algorithme général pour la résolution des systèmes linéaires. L'inverse à gauche de la matrice de Krylov dépend de deux matrices. Le choix de ces deux matrices et le polynôme $Q_k$ donne les différentes variétés des méthodes de Krylov. Pour un choix particulier des deux matrices, on a pu améliorer la convergence de quelques méthodes de Krylov. En effet, on a appliqué des projecteurs orthogonaux aux vecteurs résidus pour minimiser leurs normes et améliorer la précision de calcul par la suite. De plus, pour éviter les problèmes de stockage et garder le même stockage, on a pensé à utiliser pour chaque méthode tous les vecteurs calculer à chaque itération pour construire ces projecteurs orthogonaux. D'autre part, on a étudié une autre famille de méthodes itératives pour la résolution des systèmes linéaires appelés IDR. On a montré que l'approche des méthodes IDR est différente de celle des méthodes Krylov. Alors, on ne peut l'inclure dans

notre approche. Ceci n'était pas un inconvénient pour nous car on a pu améliorer sa convergence au niveau de la précision et la stabilité pour atteindre la précision de la méthode la plus optimal GMRES en gardant le fait que la méthode IDR est mieux au niveau du stockage et au niveau de temps. L'amélioration de la convergence se voit clairement lorsque le nombre de vecteurs utilisés pour construire le projecteur orthogonal est grand. Pour ceci, on a considéré les versions par bloc de quelques méthodes de Krylov et on a appliqué la même technique. Pour illustrer la performance des méthodes dérivées, on a considéré la fameuse méthode Bl-GMRES en tant que la méthode la plus optimale. On a montré numériquement qu'on peut atteindre la précision et la stabilité de la méthode Bl-GMRES.

# List of Algorithms

# Publications

- F. BOUYGHF, A. MESSAOUDI AND H. SADOK, *An enhancement of the convergence of the IDR method,* Electronic Transactions on Numerical Analysis Journal, 58 (2023), pp. 470–485.
- F. BOUYGHF, *An enhancement of the convergence of the BiCGStab, the Gl-BiCGStab and Bl-BiCGStab methods,* Computational and Mathematical Methods Journal, , 2023 (2023), 9 pages.
- F. BOUYGHF, A. MESSAOUDI AND H. SADOK, *A unified approach to Krylov subspace methods for solving linear systems,* Accepted for publication in Numerical Algorithms Journal.
- F. BOUYGHF, A. MESSAOUDI AND H. SADOK, *Avoiding breakdown in the Bl-MINRES method as a Bl-Lanczos type method,* In progress.

# List of talks

- **Doctoral Day** F. BOUYGHF, A. MESSAOUDI AND H. SADOK, *Characterization of the left inverse and applications,* Doctoral Day, 25-26 December 2019, ENS, Rabat, Morocco.

- **13th JANO** F. BOUYGHF, A. MESSAOUDI AND H. SADOK, *An enhancement of the convergence of IDR(s) methods,* 13th Numerical analysis and optimisation days, 22-24 February 2021, ENSA, Khouribga, Morocco.

- **JST** F. BOUYGHF, A. MESSAOUDI AND H. SADOK, *An enhancement of the convergence of the Krylov subspace methods for solving linear systems,* Scientific and technical days, University of Hassan II, 22-25 March 2021, Casablanca, Morocco.

- **9th ICAAMM** F. BOUYGHF, A. MESSAOUDI AND H. SADOK, *A new implementation of the Bl-QMR method for solving non-symmetric linear systems,* 9th International Conference on Applied Analysis and Mathematical Modelling, University of Biruni , 11-13 June 2021, Istanbul, Turkey.

- **Workshop** F. BOUYGHF, A. MESSAOUDI AND H. SADOK, *Krylov subspace methods for solving matrix equations,* Doctoral day, University of Littoral Cote d'Opale, 07 July 2021, Calais, France.

- **ICDEMMCA** F. BOUYGHF, A. MESSAOUDI AND H. SADOK, *The Block MINRES with breakdowns method for solving symmetric linear systems with multiple right-hand sides,* Differential Equations, Mathematical Modelling and Computational Algorithms, Belgorod University, 25-29 October 2021, Russia.

- **2th IMSRC** F. BOUYGHF, A. MESSAOUDI AND H. SADOK, *Convergence improvement of the induced dimension reduction methods for solving linear systems,* 2th International Modern Scientific Research Congress, 23-25 December 2021 Istanbul, Turkey.

- **ISAME 2022** F. BOUYGHF, A. MESSAOUDI AND H. SADOK, *On the Block MINRES method: From theory to implementation,* International Symposium on applied mathematics and engineering, 21-23 January 2022, University of Biruni, Istanbul, Turkey.

- **JNCF 2022** F. BOUYGHF, AN ENHANCEMENT OF THE CONVERGENCE OF IDR(S) METHOD AS A PETROV-GALERKIN TYPE METHOD Francophone computer algebra days, 28 February- 4 March 2022, University of Luminy, Marseille, France.

- **ISAME 2022** F. BOUYGHF, A. MESSAOUDI AND H. SADOK, *On the Block MINRES method: From theory to implementation,* International Symposium on Applied Mathematics and Engineering, 21-23 janvier 2022, University of Biruni , Istanbul, Turkey.

- **NMLSP 2022** F. BOUYGHF, A. MESSAOUDI AND H. SADOK, *An enhancement of the convergence of IDR method,* Numercal methods for solving large problems, 6-10 juin 2022, Faculty of Mechanical Engineering, Belgrade, Serbia.

Liste des tableaux

# Bibliography

[1] W. E. ARNOLDI, *The principe of minimized iterations in the solution of the matrix eigenvalue problem*, Quart. Appl. Math., vol. 9(1951), 17-29.

[2] E. H. AYACHOUR, *A fast implementation for GMRES method*,, J. comput. Appl. Math., vol. 159 (1986), 269-283.

[3] S. AXLER, *Linear algebra done right*, Springer Science Business Media, (1997).

[4] M. A. AKGÜN, J. H. GARCELON AND R. T. HAFTKA, *Fast exact linear and nonlinear structural reanlysis and the Sherman-Morrison-Woodbury formulas,* J. Numer. Meth. Engng., 50 (2001) 1587–1606.

[5] K. AIHARA, K. ABE AND E. ISHIWATA, *An alternative implementation of the IDRStab method saving vector updates,* J. SIAM, 3 (2011) 69–72.

[6] R. ASTUDILLO AND M. B. VAN GIJZEN, *A restarted Induced Dimension Reduction method to approximate eigenpairs of large unsymmetric matrices*, J. Comp. Appl. Math., 296 (2016) 24-35.

[7] C. BREZINSKI, M. REDIVO-ZAGLIA AND H. SADOK, *Avoiding breakdown and near breakdown in Lanczos-type algorithms*, Numer. Algorithms, 1 (1991), pp. 199–206.

[8] C. BREZINSKI, M. REDIVO-ZAGLIA AND H. SADOK, *The matrix and polynomial approaches to Lanczos-type algorithms*, Numer. Math., 45 (1992), pp. 361–376.

[9] C. BREZINSKI, M. REDIVO-ZAGLIA AND H. SADOK, *The matrix and polynomial approaches to Lanczos-type algorithms*, J. Comput. Appl. Math., 123 (2000), pp. 241–260.

[10] P.N. BROWN, *A theoretical comparison of the Arnoldi and GMRES algorithms,* SIAM J. Sci. Statist. Comput. 12 (1991) 58-78.

[11] C. BREZINSKI AND H. SADOK, *Lanczos type methods for systems of linear equations*, J. Appl. Numer. Math., 11 (1993), 443-473.

[12] C. BREZINSKI, *The block Lanczos and Vorobyev*, C.R. Acad. Sci. Paris, Série, J 331(2000), 137-142.

[13] A. BEN-ISRAEL AND T.N.E. GREVILLE, *Generalized Inverse, Theory and Application*, Second edition, Canadian Mathematical Society (2003).

[14] F. BOUYOULI, K. JBILOU, R. SADAKA, AND H. SADOK, *Convergence properties of some block Krylov subspace methods for multiple linear systems*, J. Comput. Appl. Math., 196 (2006), 498-511.

[15] J. BAGLAMA AND L. REICHEL, *Augmented GMRES type methods,* J. Numer. Linear. Algebra. Appl., 14 (2007) 337-350.

[16] B. BECKERMANN AND L. REICHEL, *The Arnoldi process and GMRES for nearly symmetric matrices, SIAM*, J. Anal. Appl., 30 (2008) 102-120.

[17] M. BELLALIJ, Y. SAAD AND H. SADOK, *Further analysis of the Arnoldi process for eigenvalue problems,* SIAM. Numer. Anal., 48 (2010) 393-407.

[18] M. BENHAMADOU, *On the FOM algorithm for the resolution of the linear systems $Ax = b$,* Linear Algebra and Matrix Theory, 4(2014) 156-171.

[19] M. BELLALIJ, G. MEURANT AND H. SADOK, *The distance of an eigenvector to a Krylov subspace and the convergence of the Arnoldi method for eigenvalue problem,* Linear. Algebra. App. 504 (2016) 387-405.

[20] M. BELLALIJ, Y. SAAD AND H. SADOK, *On the convergence of the Arnoldi process for eigenvalue problems.*

[21] F. BOUYGHF, A. MESSAOUDI, H. SADOK, *An unified approach to Krylov subspace methods for solving linear systems*, Numerical Algorithms Journal, https://doi.org/10.1007/s11075-023-01648-0.

[22] F. BOUYGHF *An enhancement ofSome techniques have already been proposed to avoid breakdowns in the conjugate gradient method the convergence of the BiCGStab method for solving linear systems with single or multiple right-hand sides*, Computational and Mathematical Methods, 2023 (2023), 9 pages.

[23] F. BOUYGHF, A. MESSAOUDI AND H. SADOK *An enhancement of the convergence of IDR method*, Electron. Trans. Numer. Anal., 58 (2023), pp. 470–485.

[24] E. CARSON, M. ROZLOZNIK, Z. STRAKOS, P. TICHY AND M. TUMA,*The numerical stability analysis of piplined conjugate gradient methods: Historical context and methodology* SIAM, Sci. Comput., 21 (1999) 1263-1290.

[25] E. CARSON, N. KNIGHT AND J. DEMMEL, *Avoiding communication in non-symmetric Lanczos based Krylov subspace methods,* SIAM, Sci. Comput. 35 (2013) S42-S61.

[26] J. CHEN, L. C. MCINNES AND H. ZHANG, *Analysis and particular use of flexible BiCGStab,* Sci. Comput., 68 (2016) 803-825.

[27] H. DAÏ, *Two algorithms for symmetric linear systems with multiple right-hand sides*, Num. Math. Theory. Meth. App. (2000).

[28] L. DU, T. SOGABE, S. L. ZHANG, *A variant of the IDR(s) method with the quasi-minimal residual strategy,* Comput. App. Math., Vol. 236 (2011), 621-630.

[29] L. DU, T. SOGABE, B. YU, Y. YAMAMOTO AND S.-L. ZHANG, *A block IDR method for non-symmetric linear systems with multiple right-hand sides*, J. Comput. App. Math., 235 (2011) 4095-4104.

[30] J. DUINTJER TEBBENS, G. MEURANT, H. SADOK AND Z. STRAKOS, *On investigating GMRES convergence using unitary matrices,* Linear Algebra App. 450 (2014) 83-107.

[31] S. DUMINIL, M. HEYOUNI, P. MARION AND H. SADOK, *Algorithm for the CMRH method for dense linear systems,* Numer. Alg., 71 (2016) 383–394.

[32] M. EL HEYOUNI, *The global Hessenberg and global CMRH methods for linear systems with multiple right-hand sides*, Numer. Algorithms, 26 (2001), 317-332.

[33] A. EL GUENNOUNI, K. JBILOU AND H. SADOK, *A block version of BiCGStab for linear systems with multiple right hand sides,* Electron. Trans. Numer. Anal., 16 (2003), 129-142.

[34] A. EL GUNNOUN, K. JBILOU AND H. SADOK, *The block Lanczos method for linear systems with multiple right-hand sides* J. Appl. Numer. Math., 51 (2004) 243-256.

[35] L. ELBOUYAHYAOUI, A. MESSAOUDI, AND H. SADOK, *Algebraic propreties of the block GMRES and Block Arnoldi methods,* Electronic Transactions on Numerical Analysis,33 (2009) 207-220.

[36] L. EDSBERG, *Introduction to computation and modeling for different equations,* second edition, Library of congress cataloging -in-publication data, (2016).

[37] R. FLETCHER, *Conjugate gradient methods for indefinite systems,* In G. A. Watson, editor, Proceedings of the Dundee Biennal Conference on Numerical Analysis (1974) 73-89. Springer Verlag, New York.

[38] R. W. FREUND AND N. M. NACHTIGAL, *QMR: a quasi-minimal residual method for non-Hermitian linear systems*, J. Numer. Math., 60 (1991) 315-339.

[39] R. W. FREUND AND M. MALHOTRA, *A bloc algorithm for non hermitian linear systems with multiple right-hand sides*, Linear Algebra Appl. 254 (1997), 119-157.

[40] FRANCISREK A. DUL, *MINRES and MINRR are better than SYMMLQ in eigenpair computations* SIAM, Sci., Comput., 19 (1998) 1767-1782.

[41] T.N.E. GREVILLE, *Some applications of the pseudoinverse of a matrix*, SIAM Review, 2 (1960), pp.15–22.

[42] M. B. V. GIJZEN, C. B. VREUGDENHIL AND H. OKSUZOGLU, *The finite element discretization for Stream-Function problems on multiply connected domains,* J. Comp. Phys., 140 (1998) 30-46.

[43] P. R. GARES-MORRIS, *The breakdowns of BiCGStab,* Num. Alg., 29 (2002) 97-105.

[44] M. H. GUTKNECHT, *Block Krylov space methods for linear systems with multiple right hand sides: an introduction,* in: Modern mathematical models, methods and algorithms for real world systems, Anamaya Publishers, New Delhi, India, 2006.

[45] M. H. GUTKNECHT, T. SCHMELZER., *Updating the QR-decomposition of block tridiagonal and block Hessemberg matrices*, Appl. Numer. Math., vol. 58 (2008), 871-83.

[46] M. B. V. GIJZEN AND P. SONNEVELD, *An elegant IDR variant that efficiently exploits bi-orthogonality properties,* Report 08-21, Department of Applied Mathematical Analysis, Delft University of Technology, (2008).

[47] M. H. GUTKNECHT AND T. SCHMELZER, *The block grade of a block Krylov space,* Linear Algebra, 430 (2009) 174-185.

[48] M. H. GUTKNECHT, *IDR Explained,* J. Elect. Trans. Num. Analys., Vol. 36 (2010) 126-148.

[49] M. B. V. GIZEN, G. L. G. SLEIJPEN, AND J-P. M. ZEMKE, *Flexible and multi-shift induced dimension reduction algorithms for solving large sparse linear systems*, Numer. Linear. Algebra. Appl., 22 (2015) 1-25.

[50] M. R. HESTENES AND E. STIEFEL, *Methods of conjugate gradients for solving linear systems,* J. Research Nat. Bur. Standards, 49 (1952) 409-436.

[51] A.S. HOUSEHOLDER AND F.L. BAUER, *On certain methods for expanding the characteristic polynomial*, Numer. Math., 1 (1959), pp. 29–37.

[52] M. HEYOUNI AND H. SADOK, *On a variable smoothing procedure for Krylov subspaces methods*, Linear Algebra Appl. 268 (1998) 131–149.

[53] T. J. R. HUGHES,*The finite element method: Linear static and dynamic finite element analysis,* Library of congress cataloging-in-publication data (2000).

[54] W. JOUBERT, *Lanczos methods for the solution of nonsymmetric systems of linear equations*, SIAM, Matrix Anal. Appl., 13 (1992) 926-943.

[55] K. JBILOU, A. MESSAOUDI, AND H. SADOK, *Global FOM and GMRES algorithms for matrix equations*, Appl. Numer. Math., 31 (1999), 49-63.

[56] K. JBILOU, A. MESSAOUDI AND K. TABA, *Some Schur complement identities and applications to matrix extrapolation methods*, Linear Algebra and its Applications, 392 (2004) 195-210

[57] K. JBILOU, H. SADOK AND A. TINZEFTE, *Oblique projection methods for linear systems with multiple right-hand sides*, Electron. Trans. Numer. Anal., 20 (2005), pp. 119-138.

[58] K. JBILOU, *A Note on the Block and Seed BiCGSTAB Algorithms for Nonsymmetric Multiple Linear Systems*, Algorithms and Computings, 3 (2016) 1-13.

[59] C. LANCZOS, *An iteration method for the solution of the eigenvalue problem of linear differential and integral operators*, J. Res. Nat. Bureau standars, 45 (1950) 255-281.

[60] P. LANCASTER, *Theory of Matrix,* Academic Press, New York, 1969.

[61] J. LIESEN, Z. STRAKOS, *Krylov subspace methods principales and analysis*, Numer. math. and scientific comput., Oxford university press, first edition (2013).

[62] P. LANCASTER, *Theory of Matrices,* Academic Press, New York, 1969.

[63] J. LIESEN, Z. STRAKOS, *Krylov subspace methods principales and analysis*, Numer. math. and scientific comput., Oxford university press, first edition (2013).

[64] *Matrix Market,* Available online at http://math.nist.gov/matrix market/.

[65] A. Messaoudi, *Recursive interpolation algorithm : A formalism for solving systems of linear equations-I Direct methods*, J. Comput. Appl. Math. 76 (1996) 13-30.

[66] A. Messaoudi, *Recursive interpolation algorithm: a formalism for solving systems of linear equations-II Iterative methods*, J. Comput. Appl. Math. 76(1996)31–53.

[67] R. B. MORGAN, *Restarted block-GMRES with deflatation of eigenvalues*, Appl. Numer. Math., vol. 54 (2005), 222-236.

[68] G. MEURANT, *Estimates of thenorm of the error in solving linear systems with FOM and GMRES* , SIAM, Sci. Comput., 33 (2011) 2686-2705.

[69] G. MEURANT, *On the residual normal norm in FOM and GMRES,* SIAM, Marix. Analysis. App., (2011).

[70] M. HAJARIAN *Matrix iterative methods for solving the Sylvester-transpose and periodic Sylvester matrix equations,* Journal of the Franklin Institute 350 (2013) 3328-3341.

[71] G. MEURANT, *Necessary and sufficient conditions for GMRES complete and partial stagnation,* App. Num. Math., 75 (2014) 100-107.

[72] G. MEURANT, *The coefficients of the FOM and GMRES residual polynomials,* SIAM Matrix Analysis App., (2015).

[73] G. MEURANT AND J. TABBENS, *Krylov methods for non-symmetric linear systems: From theory to computations,* Springer series in computational mathematics, 57 (2020).

[74] D. O'LEARY, *The block conjugate gradient algorithm and related methods,* Linear Algebra Appl., 29 (1980) 293-322.

[75] N. N. Y. ONOUE AND S. FUJINO, *An overview of a family of new iterative methods based on IDR theorem and its estimation,* Proceeding of the International Multi-Conference of Engineers and Computer Scientists Hong Kong, 2 (2009) 129-234.

[76] C. C. Paige and M. A. Saunders, *Solution of sparse indefinite systems of linear equations,* SIAM J. Numer. Anal., 12 (1975) 617-629.

[77] C. Procesi, *A formal inverse to the Cayley-Hamilton theorem,* Journal of algebra 107.1 (1987) 63-74.

[78] B. N. PARLETT, *Reduction to tridiagonal form and minimal realizations*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 567–593.

[79] B. PHILIPPE AND L. REICHEL, *On the generation of Krylov subspace bases,* RR-7099, INRIA., (2009).

[80] A. QUARTERONI, R. SACCO, ET F. SALERI, *Méthodes numériques, algorithmes, analyse et applications.*, Springer-Verlag, Italia, Milano (2007).

[81] O. RENDEL, A. RIZANOLLI, J. P. M. ZEMKE, *IDR: A new generation of Krylov subspace methods,* Linear Algebra and itd Applications, (2012).

[82] P. WESSELING AND P. SONNEVELD, *Numerical experiments with a multiple grid and a preconditioned lanczos type method*, in: Lecture No res in Mathematics, vol. 771, Springer Verlag, Berlin, Heidelberg, New York, (1980), 543-562.

[83] P. SONNELEVD, *CGS, A fast Lanczos-type solver for non-systems linear systems,* Department of Mathematics and Informatics, Delft University of Technology (1984) 84-16.

[84] Y. SAAD AND M. H. SCHULTZ, *GMRES: a generalized minimal residual algorithm for solving non-symmetric linear systems*, SIAM J. Sci Stat. Comput., 7 (1986) 856-869.

[85] G. L. G. SLEIJPEN AND D. R. FOKKEMA, *BiCGStab(l) for linear equations invoking matrices with complex spectrum,* ETNA, 1 (1994) 11-32.

[86] H. SADOK, *Méthods de projection pour les systèmes linéaires et non linéaires*, Habilitation thesis, University of Lille, 1994 .

[87] V. SIMONCINI AND E. GALLOPOULOS, *Convergence properties of block GMRES and matrix polynomials,* Linear Algebra Appl., 247 (1996) 97-119.

[88] SARKAR, K. TAPAN, S. MAGDALENA, AND C. MICHAEL WICKS,*Wavelet applications in engineering electromagnetics*, Artech House, (2002).

[89] Y., *Iterative Methods for Sparse Linear System*, second edition, SIAM, (2003).

[90] G. L. G. SLEIJPEN, P. SONNEVELD AND M. B. V. GIJZEN, *BiCGStab as an induced dimension reduction method,* Technical report 08-07, department of applied mathematical analysis, delft university of technology, delft, (2008).

[91] P. SONNEVELD, M. B. VAN GIJZEN, *IDR: A family of simple and fast algorithms for solving large nonsymmetric systems of linear equations,* SIAM, Sci. Comput., 31 (2008) 1035-1062.

[92] V. SIMONCINI AND D. B. SZYLD, *Interpreting IDR as a Petrov-Glerkin method*, SIAM J. Sci. Comp., 32 (2010) 1898-1912.

[93] G. L. G. SLEIJPEN AND M. B. VAN GIJZEN, *Exploiting BiCGStab(l) strategies to induced dimension reduction,* SIAM, J. Sci. Comput., 32 (2010) 2687-2709.

[94] H. SADOK, *CMRH: a new method for solving nonsymmetric linear systems based on the Hessenberg reduction algorithm*, Numer. Alg., 20 (1999) 303-321.

[95] Y. SAAD, *Iteratives Methods for Sparse Linear System*, second edition, SIAM, (2003).

[96] H. SADOK, *Analysis of the convergence of the minimal and orthogonal residual methods* , Numer. Algo., 40 (2005) 101-115.

[97] P. SONNEVELD, M. B. VAN GIJZEN, *IDR(s): a family of simple and fast algorithms for solving large nonsymmetric systems of linear equations,* SIAM, Sci. Comput., 31 (2008) 1035-1062.

[98] G. SACCHI AND V. SIMONCINI, *A GMRES convergence analysis for localized invariant subspace ill conditioning,* SIAM, Math. Analysis. App. 40(2019) 542-563.

[99] Y. SAAD, *Iterative methods for sparse linear systems*, PWS. Publishing Company (1996).

[100] Y. SAAD, *On the Lanczos method for solving symmetric linear systems with several right-hand sides*, Math. Comp., vol. 48 (1987), 651-662.

[101] Y. SAAD AND M. SCHULTZ, *GMRES: A generalized minimal residual algorithm for solving non-symmetric linear systems* , J. Math. Comp., vol. 48 (1987), 651-662.

[102] K. M. SOODHLTER, *A block MINRES algorithm based on the band Lanczos method*, arXiv: 1301.2102v3 [math.NA] 13 May 2014.

[103] B. VITAL, *Etude de quelques méthodes de résolution de problèmes linéaires de grande taille sur multiprocesseur,* Ph.D. Thesis, Univérsité de Rennes, Rennes, France, 1990.

[104] H. A. VAN DER VORST, *BiCGStab, A fast and smoothly converging variant of BiCG for the solution of non-symmetric linear systems,* SIAM J. Sci. Statist. Comput., 13 (1992) 631-644.

[105] H. A. VAN DER VORST, *Iterative Krylov Methods for Large Linear Systems*, Cambridge, Cambridge University Press, (2003).

[106] M. B. VAN GIJZEN, P. SONNEVELD, *An elegant IDR(s) variant that efficiently exploits bi-orthogonality properties,* Report 08-21, Department of Applied Mathematical Analysis, Delft University of Technology, (2008).

[107] J.H. WILKINSON, *The algebraic Eigenvalue Problem*, Claredon Press, Oxford, England, 1965.

[108] P. WESSELING, P. SONNEVELD, *Numerical expriments with a multiple gride and a preconditioned Lanczos type method,* Dept. of Mathematics, Delft university of technology.

[109] M. C. YEUNG, T. F. CHAN, *Ml(k)Bi-CGStab: A Bi-CGStab variant based on multiple Lanczos starting vectors,* SIAM J. Sci. Comput., Vol. 21, (1999), 1263–1290.