Une approche unifiée pour les méthodes de sous-espace de Krylov pour la résolution des systèmes linéaires

F. BOUYGHF[a,b], A. MESSAOUDI[a], H. SADOK[b]

[a] *LabMIA-SI, Université Mohammed V, Maroc .*
[b] *LMPA, Université du Littoral Côte d'Opale Calais, France.*

Abstract

Dans cet article, nous présentons 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 initial x_0 et son résidu correspondant $r_0 = f - Ax_0$. Notre analyse porte sur le polynôme minimal Φ_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 à 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 la 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 type produit. Dans ce cas, le kth 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 Ψ_k est un polynôme de degré fixe ou variable. Nous examinerons des choix particuliers de Ψ_k impliquant la convergence locale, le lissage, le stockage fixe et le coût de chaque itération. Nous donnerons également une amélioration de certaines méthodes

*Corresponding author
Email addresses: fatimabouyghf3@gmail.com (F. BOUYGHF), abderrahim.messaoudi@gmail.com (A. MESSAOUDI), hasadok@gmail.com (H. SADOK)

de type produit telles que la méthode CGS. Pour illustrer la performance des algorithmes dérivés, nous donnons quelques exemples numériques.

Numerical Algorithms
https://doi.org/10.1007/s11075-023-01648-0

ORIGINAL PAPER

A unified approach to Krylov subspace methods for solving linear systems

F. Bouyghf[1,2] · A. Messaoudi[1,3] · H. Sadok[2]

Received: 24 June 2023 / Accepted: 10 August 2023

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 Φ_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 subspace methods, which are of product-type methods. In this case, the kth 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 Ψ_k is a polynomial of fixed or variable degree. We will examine particular choices of Ψ_k involving local convergence, smoothing, fixed memory, and cost for each iteration. We will also give an enhancement of some product-type methods such as CGS. To illustrate the performance of the derived algorithms, we provide some numerical examples.

Keywords Krylov subspace methods · Left inverse · Linear systems · Convergence · Enhancement

✉ H. Sadok
sadok@univ-littoral.fr

Extended author information available on the last page of the article

Published online: 16 October 2023

Mathematics Subject Classification (2010) MSC 65F · MSC 15A

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

$$Ax = f, \tag{1}$$

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 subspace 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 [27] 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 [15], in 1952, used his method for that purpose.

Owing to their numerous advantages, Krylov subspace methods were the subject of very many investigations and several algorithms for their implementation were obtained. Among them, the famous conjugate gradient algorithm of Hestenes and Stiefel [11] when the matrix is Hermitian and the bi-conjugate gradient (BiCG) algorithm of Fletcher [7] in the general case must be mentioned. For nonsymmetric systems the most widely used Krylov subspace methods are the Full Orthogonalization Method (FOM), the Generalized Minimum Residual Method (GMRES) [23], and the BiCGStab [29]. The unknown parameters can be obtained for BiCG and GMRES by imposing a Petrov-Galerkin condition (the residual r_k is orthogonal to some k-dimensional subspace). Moreover, we know that GMRES satisfy a minim residual condition. Other Krylov subspace methods constructed without imposing an explicit Petrov-Galerkin condition, but using instead a seminorm minimization are the quasi-minimal residual method (QMR) introduced in [8], which have a low storage (in general) and a constant work per iteration, and the CMRH (Changing Minimal Residual method based on the Hessenberg algorithm) introduced in [24]. BiCG, GMRES, FOM, QMR and CMRH can be implemented by using a factorization of the Krylov matrix [31, chapter 6]. FOM and GMRES use the Arnoldi algorithm, while CMRH uses the Hessenberg algorithm [17] Since BiCG and QMR use the Lanczos algorithm, we can have a possible breakdown and a near breakdown and one must use a look-ahead Lanczos algorithm for avoiding them [4–6, 8, 9, 20, 21, 28].

Let x_0 be the initial approximation of the exact solution,

$x^* = A^{-1} f$, of the system (1) and $r_0 = f - Ax_0$ be the corresponding residual. We recall the definition of the Krylov subspace [22, 30]

Definition 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 [17, 22, 30] 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 Φ_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, Φ_m represents the minimal polynomial of A for the vectorr_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 subspace methods of product-type and illustrate the algorithm of selecting the polynomial Ψ_k for using these methods efficiently. Moreover, we will introduce an improved version of the CGS method [27], 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 matrix H. The matrix I_k is the identity matrix of size k, $e_i^{(n)}$ is the ith column of the identity matrix I_n and $I_k^{(n)}$ is the matrix $I_k^{(n)} = [e_1^{(n)}, \ldots, e_k^{(n)}]$. 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.

2 Preliminary results

In this section, we will use the general left inverse of W_k [1], 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 [16, 17, 22, 30].

2.1 Characterization of Krylov subspace methods

Let

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

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

$$\Phi_m(A) r_0 = \sum_{i=0}^{m} \sigma_i A^i \, r_0 = 0, \tag{2}$$

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

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

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 (2) can be written in matrix form as

$$W_m \widetilde{c} = r_0. \tag{3}$$

It is important to remark that the system (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 [1], i.e., $W_m^L W_m = I_m$, then it has been shown that if W_m^ℓ is a particular left inverse of W_m, a general left inverse W_m^L can be given by [1]

$$W_m^L = W_m^\ell + Z_m^T \left(I_n - W_m W_m^\ell \right), \tag{4}$$

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^ℓ of the matrix W_m by

$$W_m^\ell = (Y_m^T W_m)^{-1} Y_m^T. \tag{5}$$

Therefore, by using the general and particular left inverses, the linear system (3) can be solved in two distinct ways. Thus, (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

$$r_k^K = (I - W_k W_k^L) r_0 = f - A x_k^K, \tag{6}$$

with

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

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

$$x_k^K = x_0 + K_k W_k^L r_0, \tag{7}$$

where $K_k = A^{-1} W_k$ is the Krylov matrix whose columns are $w_0, \ldots 3, w_{k-1}$. Using the relations (4), (5), and (6), we get

$$\begin{aligned} r_k^K &= (I_n - W_k W_k^L) r_0 \\ &= \left(I_n - W_k [W_k^\ell + Z_k^T (I_n - W_k W_k^\ell)]\right) r_0 \\ &= (I_n - W_k Z_k^T)(I_n - W_k W_k^\ell) r_0. \end{aligned} \tag{8}$$

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

$$\mathscr{Y}_k = span\{y_1, \ldots, y_k\}.$$

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^ℓ 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 $\mathscr{Y}_k$, that is,

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

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 1 *The kth residual in Krylov subspace methods defined by*

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

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 (Table 1).

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

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

Since FOM is a Petrov-Galerkin method, we can obtain it with the choice $Y_k = K_k$ and $Z_k = 0$. An other possibility is to choose $Y_k = W_k$ and $Z_k^T = (W_k^T A^{-T} W_k) W_k^T A^{-T}$, hence, 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, see, for example, [26]. For the usual description of FOM and of GMRES, see [22, 23].

Table 1 The choices of the matrix Y_k

Method	Condition	Choice of Y_k
FOM [22]	$K_k^T A K_k$ nonsingular	K_k
GMRES [22, 23, 30]	A nonsingular	AK_k
Hessenberg [25]	A nonsingular	$\left[e_{p1}^{(n)}, e_{p2}^{(n)}, \ldots, e_{pk}^{(n)}\right]$
Lanczos [14, 16, 22]	$Y_k^T A K_k$ nonsingular	$\left[y, A^T y, \ldots, \left(A^T\right)^{k-1} y\right]$

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*

$$A\, V_k = V_{k+1} H_{k+1,k} = V_k\, H_k + h_{k+1,k}\, v_{k+1}\, e_k^T. \tag{9}$$

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

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

and

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

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

Lemma 1 *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*

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

and

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

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

Proof The proof of the first formula is a consequence of formulas (8) and (16) in [10]. If we apply Corollary 1, page 267 of [1] to the matrix $H_{k,k+1}$, we obtain the second formula. □

A consequence of this lemma is the following theorem

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

Proof From Lemma 1, 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

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

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

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

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

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

On the other hand

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

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. □

2.3 Generalized Hessenberg algorithm

We describe now the Generalized Hessenberg method due to Hessenberg [31], and to Householder and Bauer [13]. 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 γ_1 such that $r_0 = \gamma_1 b_1$ and define γ_{k+1} and b_{k+1} such that

$$\gamma_{k+1} b_{k+1} = \left(I - K_k (Y_k^T K_k)^{-1} Y_k^T\right) A^k r_0. \tag{10}$$

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 [24], that the matrix

$H_k^h \equiv B_k^\ell A B_k$ is a square Hessenberg matrix and the generalized Hessenberg relation follows

$$AB_k = B_{k+1} H_{k+1,k}^h, \tag{11}$$

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 [24], 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 behavior of Petrov-Galerkin methods. In [12], Heyouni and Sadok proposed the Minimizing Residual Seminorm method (MRS) and define its residual r_k^M by

$$r_k^M = r_0 - \delta_1 \; B_{k+1} H_{k+1,k}^h \left(H_{k+1,k}^h\right)^\dagger e_1^{(k+1)}, \tag{12}$$

and gave a relationship between r_k^P and r_k^M (see Theorem 2 [12]).

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}^{ℓ} 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

$$\begin{aligned}\left(H_{k+1,k}^{h}\right)^{\dagger} B_{k+1}^{\ell} A B_k &= \left(H_{k+1,k}^{h}\right)^{\dagger} B_{k+1}^{\ell} B_{k+1} H_{k+1,k}^{h} \\ &= I_k.\end{aligned} \tag{13}$$

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 *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. $x_k^P - x_{k-1}^M = \delta_1\ (q_k^h, e_1)\ d_k \quad$ *and* $\quad r_k^P - r_{k-1}^M = -\delta_1\ (q_k^h, e_1)\ Ad_k$
2. $x_k^M - x_{k-1}^M = \frac{\delta_1 (q_k^h, e_1)}{1+(h_{k+1,k}^h)^2 (q_k^h, q_k^h)}\ d_k \quad$ *and* $\quad r_k^M - r_{k-1}^M = -\frac{\delta_1 (q_k^h, e_1)}{1+(h_{k+1,k}^h)^2 (q_k^h, q_k^h)}\ Ad_k,$
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 [12]. We remark that if the matrix H_k is regular, we can write the vector d_k as

$$d_k = \frac{1}{(q_k^h, q_k^h)} B_k \left(H_k^h\right)^{-1} q_k^h. \tag{14}$$

We have also the following results

Theorem 4 *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 $Y_k^T AB_k$ 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 1 and is similar to that of Theorem 2. From the last statement, we deduce that the methods can also be defined by the general formula given in Theorem 1, since we can write

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

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.

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 2 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 5 *Let us assume that the Arnoldi matrix $H_k = V_k^T AV_k$ 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 $W_k^T r_k^G = 0$.

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

$$K_k = L_k U_k, \tag{15}$$

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 the 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

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

with H^h_k being a square upper Hessenberg matrix of dimension k. We also have the following Hessenberg relation

$$\begin{aligned} A\,L_k &= L_{k+1}U_{k+1}\begin{bmatrix} 0 \\ \\ I_k \end{bmatrix} U_k^{-1} \\ &= L_{k+1}H^h_{k+1,k} = L_k\,H^h_k + h^h_{k+1,k}\,\ell_{k+1}\,e_k^T. \end{aligned} \tag{16}$$

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 ℓ_{k+1} is defined by

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

and

$$L_k^{\ell} = \begin{bmatrix} (L_k^1) \\ \\ (L_k^2) \end{bmatrix}^{\ell} = \left[\,((L_k^1))^{-1}\;\; O\,\right].$$

This equation can be rewritten as

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

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

$$H^h_{k+1,k} = \begin{bmatrix} L^\ell_k A L_k \\ \\ h^h_{k+1,k}\left(e^{(k)}_k\right)^T \end{bmatrix} = \begin{bmatrix} (L^1_k)^{-1}(I^{(n)}_k)^T A L_k \\ \\ h^h_{k+1,k}\left(e^{(k)}_k\right)^T \end{bmatrix} = \begin{bmatrix} H^h_k \\ \\ h^h_{k+1,k}\left(e^{(k)}_k\right)^T \end{bmatrix}.$$

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 $y_1 = e^{(n)}_{i_1}$. Let us assume that the indexes $i_1, \ldots, i_k$ have already been obtained and set $Y_k = [e^{(n)}_{i_1}, \ldots, e^{(n)}_{i_k}]$. To obtain i_{k+1}, we have to compute

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

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

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

$$\begin{aligned} r^{Hess}_k &= \left(I_n - AL_k(Y^T_k A L_k)^{-1} Y^T_k\right) r_0 \\ &= r_0 - (r_0)_{i_1}\, L_{k+1} H^h_{k+1,k}\left(H^h_k\right)^{-1} e^{(k)}_1. \end{aligned}$$

The CMRH method is a Minimizing Residual Seminorm methods and its kth residual is defined by

$$\begin{aligned} r^C_k &= r_0 - (r_0)_{i_1}\, L_{k+1} H^h_{k+1,k}\left(H^h_{k+1,k}\right)^\dagger e^{(k+1)}_1 \\ &= \left(I_n - AL_k\left(\left(H^h_{k+1,k}\right)^T L^\ell_{k+1} A L_k\right)^{-1}\left(H^h_{k+1,k}\right)^T L^\ell_{k+1}\right) r_0. \end{aligned}$$

We have the two orthogonality conditions

$$[e^{(n)}_{i_1}, \ldots, e^{(n)}_{i_k}]^T r^h_k = 0, \qquad \text{and} \qquad \left(H^h_{k+1,k}\right)^T L^\ell_{k+1} r^C_k = 0.$$

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

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

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 [19].

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 $V_k \equiv [v_1, \ldots, v_k]$, $\tilde{V}_k \equiv \left[\tilde{v}_1, \ldots, \tilde{v}_k\right]$, and v_{k+1} and $\tilde{v}_{k+1}$ by

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

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

where

$$H_{k+1,k}^{(1)} = \begin{bmatrix} \tilde{V}_k^T A V_k \\ \\ \delta_{k+1}\left(e_k^{(k)}\right)^T \end{bmatrix} = \begin{bmatrix} H_k^{(1)} \\ \\ \delta_{k+1}\left(e_k^{(k)}\right)^T \end{bmatrix}, \quad H_{k+1,k}^{(2)} = \begin{bmatrix} V_k^T A^T \tilde{V}_k \\ \\ \beta_{k+1}\left(e_k^{(k)}\right)^T \end{bmatrix} = \begin{bmatrix} H_k^{(2)} \\ \\ \beta_{k+1}\left(e_k^{(k)}\right)^T \end{bmatrix}.$$

Consequently, the matrices $H_k^{(1)}$ and $H_k^{(2)}$ are upper Hessenberg matrices, with $H_k^{(2)} = \left(H_k^{(1)}\right)^T$. Therefore, the matrix $H_k^{(1)}$ is tridiagonal and will be denoted by $T_k \equiv H_k^{(1)}$. We also set $T_{k+1,k} \equiv H_{k+1,k}^{(1)}$. Therefore,

$$A V_k = V_{k+1} T_{k+1,k} = V_k T_k + \delta_{k+1} v_{k+1}\left(e_k^{(k)}\right)^T \quad \text{and} \quad A^T \tilde{V}_k = \tilde{V}_k T_k^T + \beta_{k+1}\tilde{v}_{k+1}\left(e_k^{(k)}\right)^T.$$

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

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

and its iterates are defined by $x_k^B = x_0 + V_k y_k^B$. 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 $T_k^{-1}\tilde{V}_k^T r_0 = \|r_0\| \; T_k^{-1}e_1 = \delta_1 \quad T_k^{-1}e_1$. 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 2 *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* $s_1 = q_1 = \dfrac{1}{\alpha_1}$,
3. $\theta_k = \dfrac{1}{q_k^T e_k} = \alpha_k - \dfrac{\beta_k\delta_k}{\theta_{k-1}}$ *for $k \geq 2$ and $\theta_1 = \alpha_1$,*
4. $(q_k, e_1^{(k)}) = -\dfrac{\delta_k}{\theta_k}(q_{k-1}, e_1^{(k-1)})$, $(s_k, e_1^{(k)}) = -\dfrac{\beta_k}{\theta_k}(s_{k-1}, e_1^{(k-1)})$ *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 2, we deduce that the kth 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 $p_1 = \dfrac{1}{\alpha_1} v_1$. This gives us the direct version of Lanczos bi-orthogonalization algorithm.

Algorithm 1 Direct Lanczos Biorthogonalization algorithm [22, 30].

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 $k = 1, \ldots,$ 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 $v_{k+1} = u/\delta_{k+1}$
- $\beta_{k+1} = \left(\tilde{u}, b_{k+1}\right)$ and $\tilde{v}_{k+1} = \tilde{u}/\beta_{k+1}$

EndDo.

We will show now how to derive the standard form of BiCG algorithm from the Direct Lanczos Biorthogonalization algorithm. The classical implementation of the BiCG algorithm, is usually deduced by using the LDU decomposition of T_k; see, for example, [23]. We can deduce it directly also from our approach by using the fact that $p_k = V_k \, (T_k)^{-1} \, e_k^{(k)}$ and setting

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

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

1. $\tilde{p}_k^T A v_k = \tilde{v}_k^T A p_k = 1$.
2. $span\{p_1, \ldots, p_k\} = span\{v_1, \ldots, v_k\} = K_k(A, r_0)$,
3. $span\{\tilde{p}_1, \ldots, \tilde{p}_k\} = span\{\tilde{v}_1, \ldots, \tilde{v}_k\} = K_k(A^T, \tilde{r}_0)$,
4. 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

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

On the other hands, from Theorem 4, 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 2, 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

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

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 2, we get

$$(\tilde{r}_{k-1}^B, r_{k-1}^B) = \gamma_1\|r_0\|\theta_k^2(s_k, e_1)(q_k, e_1) \quad \text{and} \quad (\tilde{p}_{k-1}^B, Ap_{k-1}^B) = \gamma_1\|r_0\|\theta_k^3(s_k, e_1)(q_k, e_1).$$

Consequently,

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

Thus,

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

Since BiCG ia a Petrov-Galerkin method, its corresponding Minimizing residual seminorm is the QMR method. Then, if we denote by r_k^Q the kth residual of the QMR

Algorithm 2 BiCG method [7, 22].

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 $k = 0, 1, \ldots,$, until convergence Do:

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

EndDo.

method, we deduce from (12) that

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

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 (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, 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 5, we deduce 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

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

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 $k = 1, \ldots,$, 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 $v_{k+1} = u/\delta_{k+1}$

$\beta_{k+1} = \left(\tilde{u}, v_{k+1}\right)$ 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 Enhancement of the convergence behavior 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 [2, 3]. 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 1 with r_k^P a residual vector of a Petrov-Galerkin method, we obtain

$$r_k^K = (I_n - AK_kZ_k^T)r_k^P, \qquad \text{and} \qquad x_k^K = x_0 + K_kZ_k^T x_k^P. \tag{17}$$

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

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

Let us consider in the following, some applications.

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

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

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

$$r_k^{EB} = (I - (AP_{k,s_k})(\left(AP_{k,s_k}\right)^{\dagger})r_k^B. \tag{18}$$

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)

$$r_k^{EB} = r_k^B - Ap_{k-1}(Ap_{k-1})^{\dagger} r_k^B, \tag{19}$$

and

$$x_k^{EB} = x_k^B + p_{k-1}(Ap_{k-1})^{\dagger} r_k^B. \tag{20}$$

We consider these new vectors, we get simplest enhanced BiCG method.

3.2 Some enhanced Krylov subspaces product type methods

The Krylov subspace methods considered in this section are product-type methods in which the kth residual is expressed by $r_k = \Psi_k(A)\Phi_k(A)r_0$, where Φ_k is a polynomial of degree k such that Φ_m is the minimal polynomial of A for r_0 and Ψ_k a polynomial of

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 $k = 0, 1, \ldots,$, until convergence Do:

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

EndDo.

degree k. Sonneveld [27] 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

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

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

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

Algorithm 5 CGS method [27].

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 $k = 0, 1, \ldots,$, until convergence Do:

$$\alpha_k = (\tilde{r}_0, r_k) / (\tilde{r}_0, Ap_k)$$
$$q_k = u_k - \alpha_k Ap_k$$
$$x_{k+1} = x_k + \alpha_k(u_k + q_k)$$
$$r_{k+1} = r_k - \alpha_k A(u_k + q_k)$$
$$\beta_k = \left(\tilde{r}_0, r_{k+1}\right) / (\tilde{r}_0, r_k)$$
$$u_{k+1} = r_{k+1} + \beta_k q_k,$$
$$p_{k+1} = u_{k+1} + \beta_k(q_k + \beta_k p_k),$$

EndDo.

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, 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

$$r_{k+1}^{ECGS} = r_{k+1}^{CGS} - [Ap_k, A\,(u_k + q_k)]\,[Ap_k, A\,(u_k + q_k)]^{\dagger}\, r_{k+1}^{CGS}, \tag{21}$$

and

$$x_{k+1}^{ECGS} = x_{k+1}^{CGS} + [p_k, (u_k + q_k)]\,[Ap_k, A\,(u_k + q_k)]^{\dagger}\, r_{k+1}^{CGS}. \tag{22}$$

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,

$$r_{k+1}^{ECGS} = (I - (AP_{k,s_k})(\left(AP_{k,s_k}\right)^{\dagger})r_{k+1}^{CGS}, \tag{23}$$

where

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

In [2, 3], similar techniques was used to improve the convergence behavior of BiCGSTAB and IDR.

4 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 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 $\|r_k\|/\|b\| \leq 10^{-10}$.

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

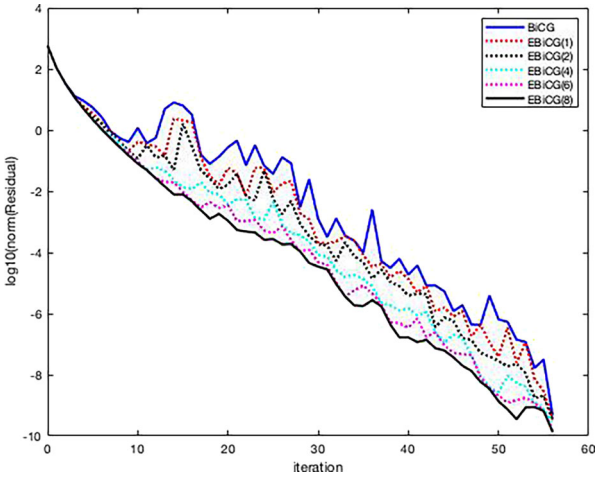

Fig. 1 Comparison between residual norms of BiCG and its enhancements EBiCG(s), for $s = 1, 2, 4, 6, 8$.

$s = 2, 4, 8$, which involve the utilization of formula (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 $s = 1, 2, 4, 6, 8$ in Figs. 1 and 2, respectively.

Figures 3 and 4 display the curves depicting the residual norms and error norms for the CGS algorithm and its enhancements. In particular, ECGS($2s$) refers to the utilization of formula (23), incorporating the available direction vectors p_{k-i}, u_{k-i} for $i = 1, \ldots, s$.

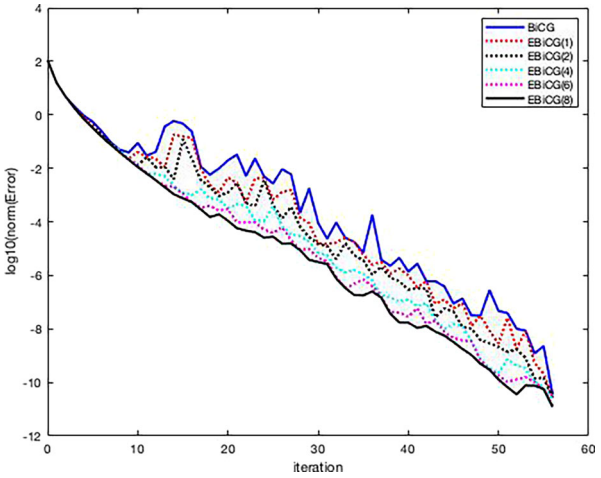

Fig. 2 The comparison between error norms of BiCG and its enhancements EBiCG(s), for $s = 1, 2, 4, 6, 8$

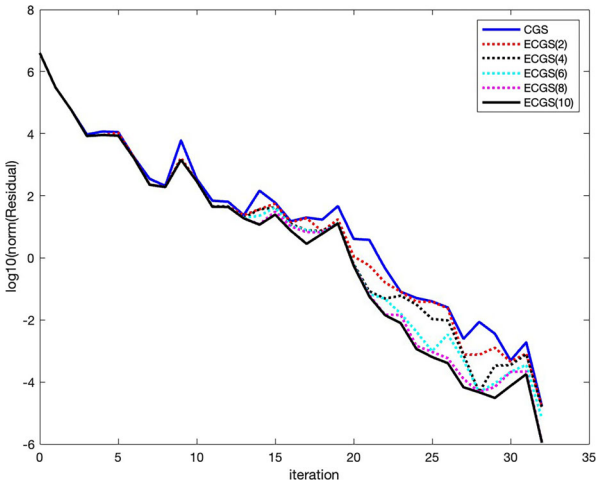

Fig. 3 The comparison between residual norms of CGS and its enhancements ECGS(s), for $s = 1, 2, 3, 4, 5$

5 Conclusion

The paper introduces a comprehensive framework for studying Krylov subspace methods used to solve linear systems of the form $Ax = f$. These methods aim to achieve convergence within a specified number of iterations, denoted as m. The focus of the analysis is on the minimal polynomial Φ_m of matrix A for the initial residual $r_0 = f - Ax_0$, with its properties playing a crucial role in the convergence behavior.

The paper establishes that Petrov-Galerkin methods and minimal seminorm methods are specific cases of Krylov subspace methods. General formulations for the iterates of these methods based on generalized inverses are presented, and their

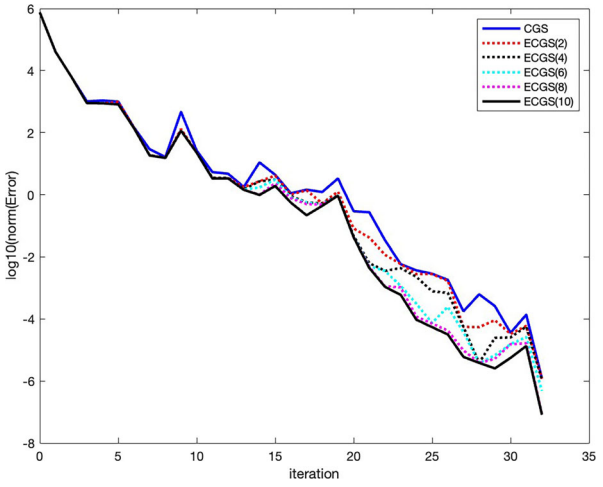

Fig. 4 The comparison between error norms of CGS and its enhancements ECGS(s), for $s = 1, 2, 3, 4, 5$

mathematical properties are analyzed. The paper also proves that specific instances of Krylov subspace methods 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. Additionally, the paper discusses some methods that are product-type methods.

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

Acknowledgements The authors would like to thank the referees for their constructive comments. Their remarks have helped us to improve the presentation of this paper.

Author contribution All three authors have contributed in the same way.

Funding None.

Data availability Data sharing not applicable to this article as no data sets were generated or analyzed during the current study. We just improved existing programs.

Declarations

Ethical approval Not applicable. External Review board.

Competing interests The authors declare no competing interests.

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Ben-Israel, A., Greville, T.N.E.: Generalized inverse, theory and application, 2nd edn. Canadian Mathematical Society, Ottawa (2003)
2. Bouyghf, F.: An enhancement of the convergence of the BiCGStab method for solving linear systems with single or multiple right-hand sides, submitted to Computational and mathematical methods
3. Bouyghf, F., Messaoudi, A., Sadok, H.: An enhancement of the convergence of IDR method. Electron. Trans. Numer. Anal. **58**, 470–485 (2023)
4. Brezinski, C., Redivo-Zaglia, M., Sadok, H.: Avoiding breakdown and near breakdown in Lanczos-type algorithms. Numer. Algorithms **1**, 199–206 (1991)
5. Brezinski, C., Redivo-Zaglia, M., Sadok, H.: The matrix and polynomial approaches to Lanczos-type algorithms. Numer. Math. **45**, 361–376 (1992)
6. Brezinski, C., Redivo-Zaglia, M., Sadok, H.: The matrix and polynomial approaches to Lanczos-type algorithms. J. Comput. Appl. Math. **123**, 241–260 (2000)
7. Fletcher, R.: Conjugate gradient methods for indefinite systems. In: Watson, G.A. (ed.) Numerical analysis, Dundee 1975. Lecture Notes in Mathematics, vol. 506. Springer, Berlin (1976)
8. Freund, R.W., Nachtigal, N.M.: QMR: a quasi minimal residual method for non-Hermitian linear systems. Numer. Math. **60**, 315–339 (1991)
9. Freund, R.W., Gutknecht, M.H., Nachtigal, N.M.: An implementation of the look-ahead Lanczos algorithm for non-Hermitian matrices. SIAM J. Sci. Comput. **14**, 137–158 (1993)
10. Greville, T.N.E.: Some applications of the pseudoinverse of a matrix. SIAM Rev. **2**, 15–22 (1960)
11. Hestenes, M.R., Stiefel, E.: Methods of conjugate gradients for solving linear systems. J. Res. Natl. Bur. Stand. **49**, 409–436 (1952)
12. Heyouni, M., Sadok, H.: On a variable smoothing procedure for Krylov subspaces methods. Linear Algebra Appl. **268**, 131–149 (1998)

13. Householder, A.S., Bauer, F.L.: On certain methods for expanding the characteristic polynomial. Numer. Math. **1**, 29–37 (1959)
14. Joubert, W.: Lanczos methods for the solution of nonsymmetric systems of linear equations. SIAM Matrix Anal. Appl. **13**, 926–943 (1992)
15. Lanczos, C.: Solution of systems of linear equations by minimized iterations. J. Res. Nat. Bur. stand. **49**, 33–53 (1952)
16. Liesen, J., Strakos, Z.: Krylov subspace methods principles and analysis, 1st edn. Oxford University Press, Oxford (2013)
17. Meurant, G., Tabbens, J.D.: Krylov methods for non-symmetric linear systems: from theory to computations. In: Springer Series in Computational Mathematics, vol. 57 (2020)
18. Nachtigal, N.M.: A look-ahead variant of the Lanczos algorithm and its application to the quasi minimal residual method for nonhermitian linear systems. Ph.D. thesis, Massachusetts Institute of Technology (1991)
19. Parlett, B.N.: Reduction to tridiagonal form and minimal realizations. SIAM J. Matrix Anal. Appl. **13**, 567–593 (1992)
20. Parlett, B.N., Taylor, D.R., Liu, Z.A.: A look-ahead Lanczos algorithm for unsymmetric matrices. Math. Comput. **44**(169), 105–124 (1985)
21. Pozza, S., Pranic, M.S.: The Gauss quadrature for general linear functionals, Lanczos algorithm, and minimal partial realization. Numer. Algorithms **88**(2), 647–678 (2021)
22. Saad, Y.: Iterative methods for sparse linear systems. The PWS Publishing Company, Boston, 1996. 2nd edition, SIAM, Philadelphia (2003)
23. Saad, Y., Schultz, M.H.: GMRES: a generalized minimal residual algorithm for solving non-symmetric linear systems. SIAM J. Sci Stat. Comput. **7**, 856–869 (1986)
24. Sadok, H.: Méthods de projection pour les systèmes linéaires et non linéaires. Habilitation thesis, University of Lille (1994)
25. Sadok, H.: CMRH: a new method for solving nonsymmetric linear systems based on the Hessenberg reduction algorithm. Numer. Algorithms **20**, 303–321 (1999)
26. Sadok, H., Szyld, D.: A new look at CMRH and its relation to GMRES. BIT Numer. Math. **52**, 485–501 (2012)
27. Sonneveld, P.: CGS: a fast Lanczos-type solver for non-symmetric linear systems. SIAM J. Sci. Stat. Comput. **10**, 36–52 (1989)
28. Taylor, D.R.: Analysis of the look ahead Lanczos algorithm. Ph.D. thesis, University of California, Berkley (1982)
29. Van Der Vorst, H.A.: Bi-CGStab: a fast and smoothy converging variant of Bi-CG for the solution of non-symmetric linear systems. SIAM J. Sci. Stat. Comput. **12**, 631–644 (1992)
30. Van der Vorst, H.A.: Iterative Krylov methods for large linear systems. Cambridge University Press, Cambridge (2003)
31. Wilkinson, J.H.: The algebraic eigenvalue problem. Claredon Press, Oxford, England (1965)

Authors and Affiliations

F. Bouyghf[1,2] · A. Messaoudi[1,3] · H. Sadok[2]

F. Bouyghf
fatimabouyghf3@gmail.com

A. Messaoudi
abderrahim.messaoudi@gmail.com

1 Laboratoire de Mathématiques, Informatique et Applications - Sécurité de l'Information (LabMiA-SI), Mohammed V University, Rabat, Morocco

2 Laboratoire de Mathématiques Pures et Appliquées, Université du Littoral Cote d'Opale, 50 Rue F. Buisson BP 699, 62228 Calais Cedex, France

3 LabMIA-SI, Mohammed V University in Rabat, Ecole Normale Supérieure, Av. Mohammed Belhassan El Ouazzani, B.P. 5118, Takaddoum, Rabat, Morocco