Numerical Algorithms
https://doi.org/10.1007/s11075-020-01007-3

ORIGINAL PAPER

On restarted and deflated block FOM and GMRES methods for sequences of shifted linear systems

Lakhdar Elbouyahyaoui[1] · Mohammed Heyouni[2] · Azita Tajaddini[3] · Farid Saberi-Movahed[4]

Received: 7 September 2019 / Accepted: 17 August 2020 / Published online: 09 September 2020

Abstract

The problem of shifted linear systems is an important and challenging issue in a number of research applications. Krylov subspace methods are effective techniques for different kinds of this problem due to their advantages in large and sparse matrix problems. In this paper, two new block projection methods based on respectively block FOM and block GMRES are introduced for solving sequences of shifted linear systems. We first express the original problem explicitly by a sequence of Sylvester matrix equations whose coefficient matrices are obtained from the shifted linear systems. Then, we show the restarted shifted block FOM (rsh-BFOM) method and derive some of its properties. We also present a framework for the restarted shifted block GMRES (rsh-BGMRES) method. In this regard, we describe two variants of rsh-BGMRES, including (1) rsh-BGMRES with an unshifted base system that applies a fixed unshifted base system and (2) rsh-BGMRES with a variable shifted base system in which the base block system can change after restart. Furthermore, we consider the use of deflation techniques for improving the performance of the rsh-BFOM and rsh-BGMRES methods. Finally, some numerical experiments are conducted to demonstrate the effectiveness of the proposed methods.

Keywords Block Arnoldi process · Block Krylov subspace methods · Deflation · Sequences of shifted linear systems

Mathematics Subject Classification (2010) MSC 65F

✉ Mohammed Heyouni
heyouni@lmpa.univ-littoral.fr

Extended author information available on the last page of the article.

1 Introduction

In this work, we consider the solution of sequences of shifted linear systems that have the form:

$$(A - \sigma_j^{(i)} I_n)\, x_j^{(i)} = b^{(i)}, \quad \text{for } i = 1, \ldots, s, \text{ and } j = 1, \ldots, k, \tag{1}$$

where $A \in \mathbb{C}^{n\times n}$, $x_j^{(i)}$, $b^{(i)} \in \mathbb{C}^n$, $\sigma_j^{(i)} \in \mathbb{C}$ and I_n is the identity matrix of size n. The coefficient matrices $A - \sigma_j^{(i)} I_n$ are assumed to be nonsingular for all $\sigma_j^{(i)}$.

The solution of linear systems having the form (1) is of great importance in a variety of practical applications and scientific fields. Shifted linear systems are encountered in several contexts such as control theory [1, 2], quantum chromodynamics problems [3–7], time-dependent partial differential equations [8–11], large-scale eigenvalue problems [12, 13], and other engineering problems [14–19]. Here, we point out that our interest in the problem (1) comes from the description of a method based on the use of the block Arnoldi process solving the multi-input Sylvester-observer equations [20].

It is clear that direct methods based on the LU decomposition are not recommended for the solution of (1) since the LU decomposition must be recalculated for each shift $\sigma_j^{(i)}$. In addition, in many practical situations, the matrix A is not explicitly available.

Usually, the coefficient matrices appearing in shifted linear systems are large and sparse, so it makes sense to use projection methods on Krylov subspaces. Thus, one of the simplest ideas for solving the ks shifted linear systems appearing in (1) is to apply some classical Krylov method separately to each system [21]. Recall that block Krylov methods are matrix free and are very attractive for solving general linear systems that have several right-hand sides but share the same coefficient matrix and which have the form:

$$A\, X = B, \tag{2}$$

where $X = [x^{(1)}, \ldots, x^{(s)}]$ and $B = [b^{(1)}, \ldots, b^{(s)}]$. Compared with Krylov methods for solving a single linear system, projection methods on block Krylov subspaces have important advantages. Indeed, the block search space allows the simultaneous computation of matrix-vector products and is much larger and contains more information than the single space. Recently, exploiting the important and well-known shift-invariance property, various and numerous Krylov subspace methods for shifted linear systems were proposed. For example, shifted versions based on the Lanczos process are described in [5, 16, 18, 22–24]. Shifted methods based on the Arnoldi process can be found in [14, 25–28] and the references therein. More recently, restarted methods that use the Hessenberg process were introduced in [29, 30]. Shifted algorithms, combined with preconditioning, deflation, and augmentation techniques, were described in [10, 15, 19, 21, 31, 32], [4, 33, 34] and [28, 35] respectively.

Denoting by Σ_j the $s \times s$ diagonal matrix whose entries are $\sigma_j^{(1)}, \sigma_j^{(2)}, \ldots, \sigma_j^{(s)}$ and letting $\mathcal{L}_j$ be the Sylvester operator defined first in [36] and also used in [37, 38] by

$$\mathcal{L}_j(X) = A\, X - X\, \Sigma_j,$$

then the family of systems in (1) can be written as

$$\mathcal{L}_j(X_j) = A\,X_j - X_j\,\Sigma_j = B, \quad \text{for} \quad j = 1, \ldots, k, \tag{3}$$

where $X_j = [x_j^{(1)}, x_j^{(2)}, \ldots, x_j^{(s)}]$. Similarly, noting that for each j, the set of shifted systems in (1) can be considered a matrix Sylvester equation, we see that it is also possible to apply a specific method, such as those developed in [37, 39, 40] and the references therein. However, and as said before, it is preferable to exploit the shift-invariance property [38, 41]. Indeed, if the initial vectors are collinear[1], then the Krylov subspaces with respect to the matrices A and the Sylvester operator $\mathcal{L}_j$ are equal for all $j \in \{1, \ldots, k\}$. The shift-invariance property makes it possible to achieve computational savings by generating only the basis for a well-chosen Krylov subspace. Thus, this computed basis will not only be used to determine the approximate solution of the selected system but will also be reused to determine cheaply the solutions of all other systems. As far as we know, what we propose here can be seen as the natural generalization to the block case of the solution of the classical shifted linear system described in the literature [23, 25, 27].

In the present paper, we focus on block Arnoldi–based methods and describe two methods for solving families of shifted linear systems of the form (1). The first of these methods is the restarted shifted block FOM method. It can be seen as a generalization to the block case of the restarted shifted FOM method introduced by Simoncini in [27]. We also describe two variants of a block generalization of the restarted shifted GMRES which was described by Frommer and Glässner in [25]. In the first of these variants, we make use of a fixed unshifted base system, while in the second variant a shifted system—which could vary at each restart—is used. We inform the reader that other generalizations of the shifted FOM and GMRES algorithms to the block case exist. However, the generalization described here is based on a block shift of the matrix A with a diagonal matrix as indicated in (3). Our generalization differs from those proposed in [19, 44–46] because the formulation of the methods mentioned in the last references is based on a scalar shift of the matrix A. In [44], Soodhalter describes a method for solving multiple shifted linear systems with unrelated right-hand sides which can be obtained from (1) by taking $k = 1$. For this, the resolution is achieved by grouping all systems into a single Sylvester equation and by using what has already been developed in this context (see, e.g., [38, 47]) to construct solvers satisfying the requirements necessary for compatibility with the subspace recycling framework (see, e.g., [48, 49] and the references therein). In addition, it should be mentioned that the author in [44] has also put forward the idea of grouping the shifts into multiple groups to deal with the task of recycling when solutions of the shifted system rely smoothly on a parameter; for more information about such applications, see Experiment 6.6 of [44]. This is applied for a different purpose than what is proposed in our present work since we have to solve k Sylvester equations having the same right-hand side, but the kernel of the idea is there and the same underlying concept is shared.

[1] When block vectors are used, some authors prefer to use the word *cospatial* instead of *collinear* [42, 43]

For reasons of memory requirements and to limit the increasing cost of storage, the shifted block FOM and GMRES methods are implemented with a restarting strategy. Unfortunately, this often leads to a significant increase in the number of matrix-vector products and this is particularly true for matrices with small eigenvalues. Indeed, it has been established that the distribution of eigenvalues of the coefficient matrices influences the convergence of Krylov subspace methods [50, 51]. On the other hand, another disadvantage of applying the restarting strategy to Krylov subspace methods is that it is actually possible that all of the information produced in the Krylov subspace can be lost at the time of restart. For this reason, augmentation or deflation techniques have been incorporated into Krylov projection methods. The basic idea of augmentation techniques is to make the search space become larger by using a subspace including further information about the solution, while deflation techniques reduce the influence of small eigenvalues [52].

We recall that in recent years, many methods based on either augmentation or deflation have been proposed [4, 19, 33–35, 45, 46, 50, 51, 53, 54]. Among them, GMRES with deflated restarting (GMRES-DR) has been proposed by Morgan [50] as a way to improve the convergence of GMRES. This method considers the harmonic Ritz vectors associated with the smallest harmonic Ritz values computed at the end of each cycle, and then, such vectors are added to the new subspace at the time of restart. In this case, the small harmonic Ritz values that have an adverse impact on the convergence are removed or deflated. It is worthwhile to mention that an extension of GMRES-DR was also proposed by Morgan [51], where the idea is to implement the block version of GMRES with deflated restarting. Following the success of GMRES-DR, Parks et al. [49] have introduced the GCRO-DR method that incorporates deflated restarting into the generalized conjugate residual with orthogonalization (GCRO) method [55].

Over the past decade, several methods based on deflation and augmentation techniques have been established for handling various types of shifted linear systems. For instance, Darnell et al. [33] have proposed GMRES-DR with shifts (GMRES-DR-Sh) in order to study the application of GMRES-DR for the solution of linear systems of equations with multiple shifts and multiple right-hand sides. By utilizing the deflated and augmented frameworks that were described by Gaul et al. [52] for Krylov subspace methods that are based on a Petrov-Galerkin condition, Soodhalter [44] has offered a recycled version of the shifted block GMRES (rsbGMRES) method for solving the family of shifted linear systems. Based on employing an effective way to obtain a proper seed block system, Sun et al. [45] have proposed a novel shifted version of BGMRES-DR for handling linear systems with multiple shifts and multiple right-hand sides, which can be also applicable in the situation of almost linearly dependent right-hand sides. In the present paper, inspired by the mechanism of the block GMRES-DR method due to Morgan [51], we also tend to incorporate the deflation technique into the restarted shifted block FOM and "the restarted shifted BGMRES with an unshifted base system" and "the restarted shifted BGMRES with a variable shifted base system."

The structure of the paper is as follows. In Section 2, results on the shift-invariance property of block Krylov subspaces are recalled. Also, the derivations

of the Block Full Orthogonalization (BFOM) and the Block Generalized Minimal Residual (BGMRES) applied to the Sylvester equation are briefly summarized. Section 3 is devoted to the description of restarted shifted block FOM method (rsh-BFOM) and its modification that include the deflation technique at the restart (rsh-BFOM-D). The restarted shifted block GMRES (rsh-BGMRES) algorithm and its deflated variant (rsh-BGMRES-D) are described in Section 4. The behavior of the described methods is illustrated by some numerical comparisons in Section 5. Finally, we draw a conclusion in Section 6.

2 Preliminaries

We begin this section with a brief review and background on block Krylov subspace methods. For more details concerning the definition of a block Krylov subspace, the notion of block orthogonality, and the inner product used within the projection procedure on a block Krylov subspace, we refer to [42, 56, 57] and the references therein.

2.1 Block Krylov subspace

Given $A \in \mathbb{C}^{n\times n}$ and $G \in \mathbb{C}^{n\times s}$, the m-th block Krylov subspace generated with the pair (A, G) and denoted by $\mathbb{K}_m(A, G)$ is defined as:

$$\mathbb{K}_m(A, G) = \text{block span}\{G, A\,G, \ldots, A^{m-1}\,G\}.$$

We note that $W \in \mathbb{C}^{n\times s}$ belongs to $\mathbb{K}_m(A, G)$ if and only if there exists $\Omega_i \in \mathbb{C}^{s\times s}$ such that $W = \sum_{i=0}^{m-1} A^i\,G\,\Omega_i$. Using the $\circ$ notation [47, 58], we also have:

$$\mathbb{K}_m(A, G) = \left\{\mathcal{P}(A) \circ G, \quad \mathcal{P} \in \mathbb{P}_{m-1,s}\right\},$$

where $\mathbb{P}_{m-1,s}$ is the set of $s \times s$ matrix-valued polynomials of degree $m-1$ and $\mathcal{P} = (\Omega_i) \in \mathbb{P}_{m,s}$ means that $\mathcal{P}(t) = \sum_{i=0}^{m} t^i\,\Omega_i$. Using the linear operator $\mathcal{L}_j$, we introduce the block subspace $\mathbb{K}(\mathcal{L}_j, G)$ spanned by the blocks $G,\ \mathcal{L}_j(G), \ldots,\ \mathcal{L}_j^{(m-1)}(G)$, i.e.,

$$\mathbb{K}_m(\mathcal{L}_j, G) = \text{block span}\left\{G, \mathcal{L}_j(G), \ldots, \mathcal{L}_j^{(m-1)}(G)\right\},$$

where $\mathcal{L}_j^{(0)}(G) = G$ and $\mathcal{L}_j^{(i)}(G) = \mathcal{L}_j(\mathcal{L}_j^{(i-1)}(G))$ [36–38].

We note that the shift-invariance property—of classical Krylov subspaces [41]—extends naturally to the block setting. More precisely, the following lemma given in [36, 37] shows that the block Krylov subspaces constructed with the matrix A and with the operator $\mathcal{L}_j$ are identical.

Lemma 1 [36] *For all $j \in \{1, \ldots, k\}$ and for all $m > 0$, we have:*

$$\mathbb{K}_m(A, G) = \mathbb{K}_m(\mathcal{L}_j, G).$$

Before moving to the description of the block Arnoldi process, we note that the previous property plays a key role in the generalization to the block case of shifted Arnoldi-based algorithms for solving sequences of shifted linear systems. Thus, if (2) and (3) need to be solved simultaneously, the previous lemma shows that it is not necessary to build two different bases to generate each of the two block spaces $\mathbb{K}_m(A, B)$ and $\mathbb{K}_m(\mathcal{L}_j, B)$.

2.2 The block Arnoldi process

The generation of a block Krylov subspace associated to A and G can be carried out by means of the block Arnoldi process described by Algorithm 1 and which constructs an orthonormal basis $\{V_1, V_2, \ldots, V_m\}$ [57].

Algorithm 1 The block Arnoldi process.

Input: $A \in \mathbb{C}^{n\times n}$, $G \in \mathbb{C}^{n\times s}$, $m \in \mathbb{N}$.

Compute the reduced QR decomposition of G, i.e., $G = V_1 H_{1,0}$;
for $j = 1, \ldots, m$ **do**
  $W = A V_j$;
  for $i = 1, \ldots, j$ **do**
    $H_{i,j} = V_i^H W$,
    $W = W - V_i H_{i,j}$;
  end for
  Compute the QR decomposition of W, i.e., $W = V_{j+1} H_{j+1,j}$;
end for

Assuming that all the $s \times s$ triangular sub-matrices $H_{j+1,j}$ are not rank deficient, then m steps of the block Arnoldi process can carried out. In this case, the following recursions hold:

$$A\,\mathbb{V}_m = \mathbb{V}_{m+1}\,\widetilde{\mathbb{H}}_m \tag{4}$$

$$= \mathbb{V}_m\,\mathbb{H}_m + V_{m+1}\,H_{m+1,m}\,E_m^{(m)^T}, \tag{5}$$

where $\mathbb{V}_m = [V_1, V_2, \ldots, V_m]$ is a $n \times ms$ orthonormal matrix and $\widetilde{\mathbb{H}}_m = \left[H_{i,j}\right]$ is $(m+1)s \times ms$ upper block Hessenberg matrix and $E_m^{(m)} = [0_s, \ldots, 0_s, I_s]^T$ is the m-th block formed by the last s columns of the identity matrix I_{ms}.

Now, we recall some properties that will be used in the next sections and whose verification is straightforward.

Proposition 1 *Suppose that m steps of Algorithm 1 are applied to the pair* (A, G) *without encountering any breakdown, then the following properties hold for every* $j \in \{1, \ldots, k\}$*:*

1. *The columns of* $\mathbb{V}_m$ *form an orthonormal basis of* $\mathbb{K}_m(\mathcal{L}_j, V_1)$.
2. *For* $Y \in \mathbb{C}^{ms\times s}$*, we have:*

$$\mathcal{L}_j\,(\mathbb{V}_m\,Y) = \mathbb{V}_{m+1}\,\overline{\mathcal{L}}_j(Y), \tag{6}$$

where

$$\overline{\mathcal{L}}_j(Y) = \widetilde{\mathbb{H}}_m\, Y - \begin{bmatrix} Y \\ 0 \end{bmatrix} \Sigma_j.$$

Proof The first item of Proposition 1 follows immediately thanks to the shift-invariance property. For the second item (6), we have:

$$\begin{aligned} \mathcal{L}_j(\mathbb{V}_m\, Y) &= A\, \mathbb{V}_m\, Y - \mathbb{V}_m\, Y\, \Sigma_j \\ &= \mathbb{V}_{m+1}\, \widetilde{\mathbb{H}}_m\, Y - \mathbb{V}_m\, Y\, \Sigma_j \\ &= \mathbb{V}_{m+1}\, \widetilde{\mathbb{H}}_m\, Y - \mathbb{V}_{m+1} \begin{bmatrix} Y \\ 0 \end{bmatrix} \Sigma_j \\ &= \mathbb{V}_{m+1}\, \overline{\mathcal{L}}_j(Y). \end{aligned}$$

□

2.3 Restarted block FOM and block GMRES methods for Sylvester equations

Here, we briefly describe the restarted BFOM and restarted BGMRES for solving multiple linear equations of the form:

$$\mathbb{L}(X) = B, \tag{7}$$

where $\mathbb{L}$ is the linear operator defined by

$$\mathbb{L}(X) = A\, X - X\, \Gamma. \tag{8}$$

and Γ is a given matrix in $\mathbb{C}^{s\times s}$.

Let $X_0 \in \mathbb{C}^{n\times s}$ be an initial block guess and $R_0 = B - \mathbb{L}(X_0)$ its corresponding initial block residual. Then, according to Lemma 1, the subspaces $\mathbb{K}_m(A, R_0)$ and $\mathbb{K}_m(\mathbb{L}, R_0)$ are the same. Thus, we seek for approximate solutions under the form $X_m = X_0 + \mathbb{V}_m\, Y_m$, where $\mathbb{V}_m$ is the orthonormal basis constructed by the block Arnoldi process applied to the pair (A, R_0) and the correction $Y_m \in \mathbb{C}^{ms\times s}$ is obtained by imposing an orthogonality condition on the associated residual $R_m = B - \mathbb{L}(X_m)$. More precisely:

– Imposing the Petrov-Galerkin condition

$$R_m \perp \mathbb{K}_m(A, R_0)$$

defines the BFOM iterates. In this case, Y_m is the solution of the $ms \times ms$ block linear system:

$$\mathbb{H}_m\, Y_m - Y_m\, \Gamma = E_1^{(m)}\, H_{1,0},$$

where $E_1^{(m)} = [I_s, 0_s, \ldots, 0_s]^T \in \mathbb{C}^{ms\times s}$ is the first s-block of the identity matrix I_{ms}.

– Imposing the minimization condition with respect to the Frobenius inner product

$$\|B - \mathbb{L}(X_m)\|_F = \min_{Z \in X_0 + \mathbb{K}_m(A, \mathbb{R}_0)} \|B - \mathbb{L}(Z)\|_F$$

yields to the BGMRES iterates. In this case, Y_m is the solution of the reduced $(m+1)s \times ms$ least-squares problem:

$$\min_{Y \in \mathbb{C}^{ms \times s}} \|E_1^{(m+1)} H_{1,0} - (\widetilde{\mathbb{H}}_m Y - Y \Gamma)\|_F. \tag{9}$$

Note that for each of the described methods, the computational and storage requirements grow with iterations. Thus, in practice, these constraints limit the value of m and cause both methods to be implemented with a restarting strategy. This assumes that if the obtained solution X_m is not sufficiently accurate, we restart the method by considering $X_0 = X_m$ as a new starting guess. Next, we describe briefly the restarted BFOM and BGMRES algorithms which are denoted by BFOM(m) and BGMRES(m), respectively.

Algorithm 2 Restarted BFOM and BGMRES methods for $\mathbb{L}(X) = B$.

Input: $A \in \mathbb{C}^{n \times n}$, $B = [b^{(1)}, \ldots, b^{(s)}] \in \mathbb{C}^{n \times s}$, $m \in \mathbb{N}$; and a tolerance ε;

1: Choose $X_0 \in \mathbb{C}^{n \times s}$; compute $R_0 = B - \mathbb{L}(X_0)$;
2: Compute the reduced QR decomposition of R_0, i.e., $R_0 = V_1 H_{1,0}$;
3: Generate $\mathbb{V}_{m+1}, \widetilde{\mathbb{H}}_m$ by Algorithm 1 applied to the pair (A, V_1);
4: Solve $\begin{cases} \mathbb{H}_m Y_m - Y_m \Gamma = E_1^{(m)} H_{1,0} & \text{(BFOM)}; \\ Y_m = \arg\min_{Y \in \mathbb{C}^{ms \times s}} \|E_1^{(m+1)} H_{1,0} - (\widetilde{\mathbb{H}}_m Y - Y \Gamma)\|_F & \text{(BGMRES)}; \end{cases}$
5: Compute the approximate solution $X_m = X_0 + \mathbb{V}_m Y_m$; $R_m = B - \mathbb{L}(X_m)$;
6: **if** $\|R_m\|_F < \varepsilon$ **then**
7: stop;
8: **else**
9: update $X_0 \leftarrow X_m$; $R_0 \leftarrow R_m$; goto line 2;
10: **end if**

We end this section by pointing that if $\Gamma = 0_{s \times s}$, then solving (7) is equivalent to solve the block linear system (2). Consequently, Algorithm 2 coincides with the classical block FOM and block GMRES methods [47, 57, 59].

3 The restarted shifted block FOM method with deflation

This section focuses on deriving a block version for the restarted shifted FOM described in [27] in order to solve the family of shifted linear systems (3). To retain the convergence rate of the restarted shifted BFOM method, we incorporate a deflation technique that was initially proposed by Morgan in [51]. This procedure computes—at each restart—the eigenvectors associated to the smallest eigenvalues and incorporate them in the projection process to suppress the influence of the small eigenvalues.

3.1 The restarted shifted block FOM method

Let $\mathbb{V}_m$ and $\mathbb{H}_m$ be the orthonormal basis and the upper block Hessenberg matrix generated by the block Arnoldi process applied to (A, B) respectively. For each $j = 1, \ldots, k$, starting in the first cycle from an initial guess $X_0^{(j)} = 0_{n\times s}$, we seek for an approximate solution $X_m^{(j)}$ such that:

$$X_m^{(j)} = \mathbb{V}_m\, Y_m^{(j)}.$$

Denoting by $R_m^{(j)} = B - \mathcal{L}_j(X_m^{(j)})$ the associated residual, then according to the Petrov-Galerkin condition $R_m^{(j)} \perp \mathbb{K}_m(A, B)$, the correction $Y_m^{(j)}$ is obtained as the solution of the reduced shifted block system:

$$\mathbb{H}_m\, Y_m^{(j)} - Y_m^{(j)}\, \Sigma_j = E_1^{(m)}\, H_{1,0}.$$

Note that the previous equation can be seen as a particular Sylvester equation since Σ_j is a diagonal matrix. We also mention that, thanks to the shift-invariance property, we see that once a basis has been generated for one of the block shifted linear systems (3), it could also be reused for all other block shifted linear systems. So, the expensive cost required for constructing an orthonormal basis for each Krylov subspace $\mathbb{K}_m(\mathcal{L}_j, R_0^{(j)})$ is reduced. Using (5) and (6), we can express for $j \in \{1, \ldots, k\}$ the residual $R_m^{(j)}$ as follows:

$$R_m^{(j)} = -V_{m+1}\, H_{m+1,m}\, {E_m^{(m)}}^T\, Y_m^{(j)} = -V_{m+1}\, H_{m+1,m}\, (Y_m^{(j)})_m,$$

where $(Y_m^{(j)})_m$ is the m-th $s \times s$ sub-matrix of $Y_m^{(j)}$. This shows that all residuals produced at the m-th step by the shifted BFOM are collinear with the $(m + 1)$-th block V_{m+1}. These observations are summarized in the following proposition.

Proposition 2 *For each $j = 1, \ldots, k$, let $X_m^{(j)} = \mathbb{V}_m\, Y_m^{(j)}$ be an approximate solution to (3) where $\mathbb{V}_m$ is the block orthonormal basis constructed by Algorithm 1 applied to the pair (A, B). Then, the residuals $R_m^{(j)}$ are all collinear with V_{m+1}. More precisely, there exists $\beta_m^{(j)} = -H_{m+1,m}\, (Y_m^{(j)})_m \in \mathbb{C}^{s\times s}$ such that*

$$R_m^{(j)} = V_{m+1}\, \beta_m^{(j)}.$$

Similarly to the shifted FOM [27], we note that the shifted BFOM can be restarted by taking any of the residuals $R_m^{(j)}$, $j = 1, \ldots, k$ as a new initial block residual for constructing a new block Krylov subspace.

At the end of the first cycle, we consider a new starting block $\bar{V}_1$ to build a new basis $\bar{\mathbb{V}}_m$ by taking $\bar{V}_1 = V_{m+1}$. In this case, we have $\bar{V}_1 = R_m^{(j)}\, (\beta_m^{(j)})^{-1}$, for $j = 1, \ldots, k$. This allows us to still apply the Petrov-Galerkin condition with respect to the new block Krylov subspace. Hence, at the end of the cycle and for the j-th block shifted system, the corresponding new approximate solution $\bar{X}_m^{(j)}$ is updated by

$\bar{X}_m^{(j)} = X_m^{(j)} + \bar{\mathbb{V}}_m \bar{Y}_m^{(j)}$, for $j = 1, \ldots, k$, and where $\bar{Y}_m^{(j)}$ is solution of the projected shifted block system:

$$\bar{\mathbb{H}}_m \bar{Y}_m^{(j)} - \bar{Y}_m^{(j)} \Sigma_j = E_1^{(m)} \beta_m^{(j)}, \quad \text{for } j = 1, \ldots, k. \tag{10}$$

Similarly, from the previous description, it is clear that for each shifted block system we again have:

$$\bar{R}_m^{(j)} = R_m^{(j)} - (A \bar{\mathbb{V}}_m \bar{Y}_m^{(j)} - \bar{\mathbb{V}}_m \bar{Y}_m^{(j)} \Sigma_j) = -\bar{V}_{m+1} \bar{H}_{m+1,m} E_m^{(m)^T} \bar{Y}_m^{(j)} = -\bar{V}_{m+1} \bar{H}_{m+1,m} (\bar{Y}_m^{(j)})_m.$$

We next summarize the restarted shifted block FOM method for solving (3).

Algorithm 3 Restarted shifted block FOM.

Input: $A \in \mathbb{C}^{n\times n}$, $B = [b^{(1)}, \ldots, b^{(s)}] \in \mathbb{C}^{n\times s}$, $\{\sigma_j^{(i)}\}_{j=1,\ldots,k}^{i=1,\ldots,s} \subset \mathbb{C}$; $m \in \mathbb{N}$;

1: Set $X_0 = 0_{n\times s}$, $R_0 = B$ and $\mathcal{J} = \{1, \ldots, k\}$;
2: Compute the reduced QR decomposition of R_0, i.e., $R_0 = V_1 H_{1,0}$;
3: **for** $j = 1, \ldots, k$ **do**
4: Set: $\Sigma_j = \text{diag}([\sigma_j^{(1)}, \ldots, \sigma_j^{(s)}])$, $X_m^{(j)} = X_0$, $\beta_m^{(j)} = H_{1,0}$;
5: **end for**
6: Generate $\mathbb{V}_{m+1}$, $\widetilde{\mathbb{H}}_m$ by Algorithm 1 applied to the pair (A, V_1);
7: **for** each $j \in \mathcal{J}$ **do**
8: solve $\mathbb{H}_m Y_m^{(j)} - Y_m^{(j)} \Sigma_j = E_1^{(m)} \beta_m^{(j)}$;
9: update $X_m^{(j)} \leftarrow X_m^{(j)} + \mathbb{V}_m Y_m^{(j)}$;
10: Set $\beta_m^{(j)} = -H_{m+1,m} (Y_m^{(j)})_m$;
11: **end for**
12: Eliminate converged systems and update $\mathcal{J}$;
13: **if** $\mathcal{J} = \emptyset$ **then**
14: exit;
15: **else**
16: update $V_1 \leftarrow V_{m+1}$ and goto line 6;
17: **end if**

Remark 1 We end this section by listing some remarks:

1. The $ms \times s$ Sylvester (10) can be solved by a standard direct method like the Hessenberg-Schur algorithm [60].
2. In line 8 of Algorithm 3, obtaining $Y_m^{(j)}$ the solutions of the k Sylvester (10) is mathematically equivalent to obtaining $\mathbb{Y}_m = [Y_m^{(1)}, \ldots, Y_m^{(k)}] \in \mathbb{C}^{ms\times ks}$ the solution of the single Sylvester equation:

$$\mathbb{H}_m \mathbb{Y}_m - \mathbb{Y}_m \Delta = \mathbb{E},$$

where $\Delta = \text{diag}(\Sigma_1, \ldots, \Sigma_k) \in \mathbb{C}^{ks\times ks}$, $\mathbb{E} = E_1^{(m)}[\beta_m^{(1)}, \ldots, \beta_m^{(k)}] \in \mathbb{C}^{ms\times ks}$. Moreover, since Δ is a diagonal matrix, then each column $y_i^{(j)}$ $(i = 1, \ldots, s)$ of each block $Y_m^{(j)}$ $(j = 1, \ldots, k)$ is the solution of the single linear system:

$$(\mathbb{H}_m - \sigma_i^{(j)} I_{ms}) y_i^{(j)} = E_1^{(m)} (\beta_m^{(j)})_i,$$

where $(\beta_m^{(j)})_i$ is the i-th column of $\beta_m^{(j)}$.

3.2 The deflated restarted shifted BFOM

It is known that, in practice, block linear solvers can suffer from a linear dependence between some columns of the residuals [51, 53, 54]. Taking inspiration from the deflation strategy which has been proposed by Morgan [51], we propose the restarted shifted block FOM method with deflation of eigenvalues (called rsh-BFOM-D) for solving the matrix (3). The main idea here is to deflate the smallest eigenvalues in magnitude to improve convergence of the rsh-BFOM method. To begin, we recall the definition of a Ritz pair which plays an important role in deflated restarting.

Definition 1 Assume that $\mathcal{U}$ is a generic subspace of $\mathbb{C}^n$. Let $C \in \mathbb{C}^{n\times n}$, $\theta \in \mathbb{C}$ and $y \in \mathcal{U}$. Then, the pair (θ, y) is a Ritz pair of C with respect to $\mathcal{U}$ if and only if it satisfies the following relation:

$$C\, y - \theta y \perp \mathcal{U},$$

or equivalently, for the canonical scalar product,

$$\forall\, w \in \text{range}(\mathcal{U}), \quad w^H\, (C\, y - \theta\, y) = 0.$$

The vector y is called a Ritz vector associated with the Ritz value θ.

Assume that one cycle of the restarted shifted BFOM method applied to the Sylvester (3) is performed and that the recursions (4) and (5) are obtained. Then, before restarting, we determine l Ritz vectors g_i associated to the l smallest eigenvalues θ_i $(i, \ldots, l)$ where l is a suitable integer. Then, in view of Definition 1, the Ritz pairs (θ_i, g_i) (for $i = 1, 2, \ldots, l$) of A in range$(\mathbb{V}_m)$ satisfy:

$$\mathbb{V}_m^H\, (A\, \mathbb{V}_m\, g_i - \theta_i\, \mathbb{V}_m\, g_i) = 0.$$

Now, since that $\mathbb{V}_m$ is orthonormal and $\mathbb{H}_m = \mathbb{V}_m^H\, A\, \mathbb{V}_m$ and taking into account the previous relation then the l targeted eigenpairs (θ_i, g_i) are obtained as the solution of the following eigenvalue problem:

$$\mathbb{H}_m\, g_i = \theta_i\, g_i, \quad i = 1, 2, \ldots, l. \tag{11}$$

Let us denote by $r_i^{Ritz} = A\, \mathbb{V}_m\, g_i - \theta_i\, \mathbb{V}_m\, g_i$ the Ritz residual vector for $i = 1, 2, \ldots, l$ and by $R_m^{(j_0)}$ the residual associated to the approximate solution $X_m^{(j_0)}$, where the index j_0 is obtained by the seed selection strategy used in [61, 62], i.e., j_0 is the index satisfying:

$$\|R_m^{(j_0)}\|_F = \max_{j=1,\ldots,k} \|R_m^{(j)}\|_F. \tag{12}$$

Next, we establish a result that gives a useful relation between the Ritz residual vectors r_i^{Ritz} and the residual $R_m^{(j_0)}$.

Lemma 2 *Assume that one cycle of the restarted shifted block FOM applied to the Sylvester (3) has been performed and let* $r_i^{Ritz} = A\, \mathbb{V}_m\, g_i - \theta_i\, \mathbb{V}_m\, g_i$, *for*

$i = 1, 2, \ldots, l$, be the Ritz residual vectors. Then, there exists a matrix $T \in \mathbb{C}^{s\times l}$ such that:

$$[r_1^{Ritz}, \ldots, r_l^{Ritz}] = R_m^{(j_0)}\, T. \tag{13}$$

Proof By using the relations (5) and (11), it follows that for $i = 1, 2, \ldots, l$

$$\begin{aligned} r_i^{Ritz} &= (\mathbb{V}_m\, \mathbb{H}_m + V_{m+1}\, H_{m+1,m}\, {E_m^{(m)}}^T)\, g_i - \theta_i\, \mathbb{V}_m\, g_i \\ &= \mathbb{V}_m\, (\mathbb{H}_m - \theta_i\, I)\, g_i + V_{m+1}\, H_{m+1,m}\, {E_m^{(m)}}^T\, g_i \\ &= V_{m+1}\, H_{m+1,m}\, {E_m^{(m)}}^T\, g_i \qquad (14) \\ &= \mathbb{V}_{m+1} \underbrace{\begin{bmatrix} 0_{ms\times 1} \\ H_{m+1,m}\, {E_m^{(m)}}^T\, g_i \end{bmatrix}}_{=:\widetilde{r}_i}, \end{aligned}$$

where $\widetilde{r}_i \in \mathbb{C}^{(m+1)s}$. As a result, we deduce that $r_i^{Ritz} \in \text{span}\{\mathbb{V}_{m+1}\}$.
On the other hand, using the relations (4) and $X_m^{(j_0)} = X_0 + \mathbb{V}_m\, Y_m^{(j_0)}$ implies that the residual $R_m^{(j_0)}$ of the selected block system can be expressed as:

$$R_m^{(j_0)} = -V_{m+1}\, H_{m+1,m}\, {E_m^{(m)}}^T\, Y_m^{(j_0)} \tag{15}$$

$$= \mathbb{V}_{m+1} \underbrace{\begin{bmatrix} 0_{ms\times s} \\ -H_{m+1,m}\, {E_m^{(m)}}^T\, Y_m^{(j_0)} \end{bmatrix}}_{=:\widehat{R}_m} \tag{16}$$

which reveals that $R_m^{(j_0)} \in \text{range}(\mathbb{V}_{m+1})$ since $\widehat{R}_m \in \mathbb{C}^{(m+1)s\times s}$.

Moreover from (14) and (15), we deduce that $r_i^{Ritz} \in \text{range}(R_m^{(j_0)})$ for $i = 1, 2, \ldots, l$, then there exist scalars $t_{ji} \in \mathbb{C}$ such that:

$$r_i^{Ritz} = \sum_{j=1}^{s} t_{j,i}\, R_m^{(j_0)}(:, j),$$

where $R_m^{(j_0)}(:, j)$ denotes the jth column of $R_m^{(j_0)}$. Finally letting $T = [t_{ji}]$, it follows that

$$[r_1^{Ritz}, \ldots, r_l^{Ritz}] = R_m^{(j_0)}\, T,$$

which completes the proof. □

Now, suppose the l targeted Ritz vectors are given by $\mathbb{V}_m\, G_l$, where $G_l = [g_1, \ldots, g_l] \in \mathbb{C}^{ms\times l}$. Then, using relations (13) and (16), we have:

$$\begin{aligned} A\, \mathbb{V}_m\, G_l &= \mathbb{V}_m\, G_l\, \text{diag}(\theta_1, \ldots, \theta_l) + \mathbb{V}_{m+1}\, \widehat{R}_m\, T \\ &= \mathbb{V}_{m+1} \begin{bmatrix} G_l\, \text{diag}(\theta_1, \ldots, \theta_l) \\ 0_{s\times l} \end{bmatrix} + \mathbb{V}_{m+1}\, \widehat{R}_m\, T \\ &= \mathbb{V}_{m+1} \begin{bmatrix} \begin{bmatrix} G_l \\ 0_{s\times l} \end{bmatrix}, \widehat{R}_m \end{bmatrix} \begin{bmatrix} \text{diag}(\theta_1, \ldots, \theta_l) \\ T \end{bmatrix}. \end{aligned}$$

Also assume that $G_l = Q_l \Gamma_l$ is the reduced QR factorization of G_l, where $Q_l \in \mathbb{C}^{ms \times l}$ is an orthonormal matrix and $\Gamma_l \in \mathbb{C}^{l \times l}$ is an upper triangular matrix. Therefore, we augment G_l by defining $G_{l+1} \in \mathbb{C}^{(m+1)s \times (l+1)s}$ to be

$$G_{l+1} = \left[\begin{bmatrix} Q_l \\ 0_{s \times l} \end{bmatrix}, \widehat{R}_m \right], \tag{17}$$

and get

$$A \, \mathbb{V}_m \, Q_l \, \Gamma_l = \mathbb{V}_{m+1} \, G_{l+1} \begin{bmatrix} \Gamma_l \, \mathrm{diag}(\theta_1, \ldots, \theta_l) \\ T \end{bmatrix}.$$

Post-multiplying the above relation by Γ_l^{-1} gives:

$$A \, \mathbb{V}_m \, Q_l = \mathbb{V}_{m+1} \, G_{l+1} \begin{bmatrix} \Gamma_l \, \mathrm{diag}(\theta_1, \ldots, \theta_l) \\ T \end{bmatrix} \Gamma_l^{-1}.$$

Thereafter, orthogonalizing the block matrix $\widehat{R}_m$ against $\begin{bmatrix} Q_l \\ 0_{s \times l} \end{bmatrix}$ gives the QR decomposition $G_{l+1} = Q_{l+1} \, \Gamma_{l+1}$ where $Q_{l+1} \in \mathbb{C}^{(m+1)s \times (s+l)}$ and $\Gamma_{l+1} \in \mathbb{C}^{(s+l) \times (s+l)}$. Thus,

$$A \, \mathbb{V}_m \, Q_l = \mathbb{V}_{m+1} \, Q_{l+1} \Gamma_{l+1} \begin{bmatrix} \Gamma_l \, \mathrm{diag}(\theta_1, \ldots, \theta_l) \\ T \end{bmatrix} \Gamma_l^{-1}.$$

Therefore, denoting by $\mathbb{V}_l^{new} = \mathbb{V}_m \, Q_l$, $\widetilde{\mathbb{H}}_l^{new} = \Gamma_{l+1} \begin{bmatrix} \Gamma_l \, \mathrm{diag}(\theta_1, \ldots, \theta_l) \\ T \end{bmatrix} \Gamma_l^{-1}$ and $\mathbb{V}_{l+s}^{new} = \mathbb{V}_{m+1} \, Q_{l+1}$, it can be readily verified that $\widetilde{\mathbb{H}}_l^{new} = Q_{l+1}^H \, \widetilde{\mathbb{H}}_m \, Q_l$, and also, we have:

$$A \, \mathbb{V}_l^{new} = \mathbb{V}_{l+s}^{new} \, \widetilde{\mathbb{H}}_l^{new},$$

which shows that the block Arnoldi relation holds.

In order to complete the current cycle of the restarted shifted block FOM method with deflation, we perform $m - \frac{l}{s}$ iterations (in which it is assumed that l is divisible by s) of the block Arnoldi process with the starting matrix $\mathbb{V}_{l+s}^{new}(:, l+1 : l+s)$.

After updating the matrices $\mathbb{V}_{m+1}$ and $\widetilde{\mathbb{H}}_m$, we continue our algorithm by solving the updated projected (10). What is explained so far is summarized in Algorithm 4.

Algorithm 4 Restarted shifted block FOM with deflation of eigenvalues (rsh-BFOM-D).

Input: $A \in \mathbb{C}^{n\times n}$, $B \in \mathbb{C}^{n\times s}$, $\{\sigma_j^{(i)}\}_{j=1,2,\ldots,k}^{i=1,2,\ldots,s}$ and $m \in \mathbb{N}$.

1: Set $X_0 = 0_{n\times s}$, $R_0 = B$ and $\mathcal{J} = \{1, 2, \ldots, k\}$;
2: Perform one cycle of the shifted block FOM described by Algorithm 3 to generate the matrices $\mathbb{V}_{m+1}$, $\widetilde{\mathbb{H}}_m$ and the approximate solutions $X_m^{(j)}$ with their corresponding residuals $R_m^{(j)}$;
3: **for** each $j \in \mathcal{J}$ **do**
4: set $X_0^{(j)} = X_m^{(j)}$ and $R_0^{(j)} = R_m^{(j)}$;
5: **end for**
6: Eliminate converged systems and update $\mathcal{J}$; If $\mathcal{J} = \emptyset$ exist;
7: Select j_0 the index of the base block system using (12).
8: Compute the eigenpairs (θ_i, g_i) of the eigenvalue problem (11); Let $g_1, \ldots, g_l$ be the l eigenvectors corresponding to the l smallest eigenvalues (If g_i is complex consider the real and complex parts of g_i as two distinct eigenvectors. In this case, it may be necessary to adjust l); set $G_l = [g_1, g_2, \ldots, g_l] \in \mathbb{C}^{ms\times l}$;
9: Compute the reduced QR factorizations of G_l and G_{l+1}, i.e., $G_l = Q_l\,\Gamma_l$, $G_{l+1} = Q_{l+1}\,\Gamma_{l+1}$; % G_{l+1} is defined by (17).
10: Set $\mathbb{V}_l^{new} = \mathbb{V}_m\,Q_l$, $\mathbb{V}_{l+s}^{new} = \mathbb{V}_{m+1}\,Q_{l+1}$ and $\widetilde{\mathbb{H}}_l^{new} = Q_{l+1}^H\,\widetilde{\mathbb{H}}_m\,Q_l$;
11: Let $\mathbb{V}_l = \mathbb{V}_l^{new}$, $\widetilde{\mathbb{H}}_l = \widetilde{\mathbb{H}}_l^{new}$ and $\mathbb{V}_{l+s} = \mathbb{V}_{l+s}^{new}$;
12: **for** $j = \frac{l}{s} + 1, \ldots, m$ **do**
13: $W = A\,V_j$;
14: **for** $i = 1, \ldots, j$ **do**
15: Set $H_{i,j} = V_i^H\,W$, $W = W - V_i\,H_{i,j}$;
16: **end for**
17: Compute the reduced QR decomposition of W, i.e., $W = V_{j+1}\,H_{j+1,j}$;
18: **end for**
19: **for** each $j \in \mathcal{J}$ **do**
20: solve $\mathbb{H}_m\,Y_m^{(j)} - Y_m^{(j)}\,\Sigma_j = \mathbb{V}_m^H\,R_0^{(j)}$;
21: set $X_m^{(j)} = X_0^{(j)} + \mathbb{V}_m\,Y_m^{(j)}$;
22: compute $R_m^{(j)} = B - (A\,X_m^{(j)} - X_m^{(j)}\,\Sigma_j)$;
23: **end for**
24: Eliminate converged systems and update $\mathcal{J}$; If $\mathcal{J} = \emptyset$ exist.
25: **for** each $j \in \mathcal{J}$ **do**
26: update $X_0^{(j)} \leftarrow X_m^{(j)}$ and $R_0^{(j)} \leftarrow R_m^{(j)}$;
27: **end for**
28: go to line 7;

4 The restarted shifted block GMRES method with deflation

In this section, we aim to describe a block version of the shifted GMRES algorithm introduced in [25] in order to solve (3). First, according to (9), the m-th residual of BGMRES applied to solve (2) is given by $R_m = \mathbb{V}_{m+1}\, Z_{m+1}$, where

$$Z_{m+1} = E_1^{(m+1)}\, H_{1,0} - \widetilde{\mathbb{H}}_m\, Y_m. \tag{18}$$

Moreover, the approximate solution X_m can be expressed as $X_m = X_0 + \mathcal{Q}_{m-1}(A) \circ R_0$, where $\mathcal{Q}_{m-1}$ is a right matrix-valued polynomial of degree $\leq m-1$ and its residual satisfies $R_m = R_0 - A\, \mathcal{Q}_{m-1}(A) \circ R_0 = \mathcal{P}_m(A) \circ R_0$, with $\mathcal{P}_m(t) = I_s - t\, \mathcal{Q}_{m-1}(t)$ and $\mathcal{P}_m(0) = I_s$. In the sequel, a similar notation will be adopted for the shifted block systems (3) for which we seek for approximate solutions $X_m^{(j)} \in X_0^{(j)} + \mathbb{K}_m(\mathcal{L}_j, R_0^{(j)})$ and which can be formulated as:

$$X_m^{(j)} = X_0^{(j)} + \mathcal{Q}_{m-1}^{(j)}(\mathcal{L}_j) \circ R_0^{(j)}, \tag{19}$$

where $\mathcal{Q}_{m-1}^{(j)}$ is a matrix valued-polynomial such that

$$\mathcal{Q}_{m-1}^{(j)}(\mathcal{L}_j) \circ R_0^{(j)} = \sum_{i=0}^{m-1} \mathcal{L}_j^{(i)}(R_0^{(j)})\, \Omega_i, \qquad \text{where } \Omega_i \in \mathbb{R}^{s\times s}.$$

The corresponding residual is given by

$$R_m^{(j)} = R_0^{(j)} - (A\, \mathcal{Q}_{m-1}^{(j)}(\mathcal{L}_j) - \mathcal{Q}_{m-1}^{(j)}(\mathcal{L}_j)\, \Sigma_j) \circ R_0^{(j)} = \mathcal{P}_m^{(j)}(\mathcal{L}_j) \circ R_0^{(j)}.$$

To derive a recursion for obtaining $X_m^{(j)}$, we will use similar ideas to those used for the shifted GMRES [25]. Hence, we assume that the starting residuals satisfy $R_0^{(j)} = R_0\, \Psi_0^{(j)}$ for each $j = 1, \ldots, k$, (it suffices to choose $X_0^{(j)} = 0$). Then, applying BGMRES to the base block system (2), we obtain an approximate solution to (3) by requiring a collinearity condition with the residual of the base block system, i.e.,

$$R_m^{(j)} = R_m\, \Psi_m^{(j)}, \tag{20}$$

where the collinearity coefficient $\Psi_m^{(j)} \in \mathbb{C}^{s\times s}$. Using matrix valued-polynomials, we verify that $\mathcal{P}_m^{(j)}(\mathcal{L}_j) \circ R_0^{(j)} = (\mathcal{P}_m(A) \circ R_0)\, \Psi_m^{(j)}$. Then, since $R_0^{(j)} = R_0 \psi_0^{(j)}$, we have:

$$\mathcal{P}_m^{(j)}(t\, I_s - \Sigma_j)\, \Psi_0^{(j)} = \mathcal{P}_m(t)\, \Psi_m^{(j)}, \quad \text{with } \mathcal{P}_m^{(j)}(0) = I_s. \tag{21}$$

From the previous discussion, we are now in a position to state the following result.

Lemma 3 *Assume that the block Krylov space $\mathbb{K}_{m+1}(A, R_0)$ is full-rank and that $\Psi_0^{(j)}$ is a nonsingular matrix. Then, there exist a matrix polynomial $\mathcal{P}_m^{(j)}$ and a matrix $\Psi_m^{(j)}$ satisfying (21) if and only if $\mathcal{P}_m(\Sigma_j)$ is nonsingular. We then have*

$$\Psi_m^{(j)} = \mathcal{P}_m(\Sigma_j)^{-1}\, \Psi_0^{(j)} \quad \text{and} \quad \mathcal{P}_m^{(j)}(t\, I_s - \Sigma_j) = \mathcal{P}_m(t)\, \mathcal{P}_m(\Sigma_j)^{-1}.$$

We mention that in practice the matrix polynomial $\mathcal{P}_m$ needs not be calculated. Moreover, the collinearity condition (20) is not always satisfied, but when it is so, the residual $R_m^{(j)}$ and the corresponding vector solution $X_m^{(j)}$ are unique.

We continue our description of the shifted BGMRES algorithm by showing how to obtain practically an approximate solution $X_m^{(j)}$ for each block shifted system (3). This description, which depends on the choice of the base system, will be detailed in the following two sub-sections.

4.1 The shifted BGMRES algorithm with an unshifted base system

Here, and as in [25], the non-shifted block system (2) is considered a base block system. It is clear that this base system is fixed and does not change after restarting. By taking $X_0^{(j)} = X_0 = 0_{n\times s}$, we have $R_0^{(j)} = R_0\,\Psi_0^{(j)} = B$ with $\Psi_0^{(j)} = I_s$. As X_m and $X_m^{(j)}$ are in $\mathbb{K}_m(A, R_0) = \mathbb{K}_m(\mathcal{L}_j, R_0^{(j)})$, then BGMRES gives an approximate solution $X_m = X_0 + \mathbb{V}_m\,Y_m$ such that its corresponding residual is $R_m = \mathbb{V}_{m+1}\,Z_{m+1}$, where Z_{m+1} is given by (18). The approximate solutions $X_m^{(j)} = X_0^{(j)} + \mathbb{V}_m\,Y_m^{(j)}$ of the shifted block system (3) can be updated by requiring the collinearity condition (20). This implies:

$$\begin{aligned} R_m^{(j)} = R_m\,\Psi_m^{(j)} &\Leftrightarrow R_0^{(j)} - (A\,\mathbb{V}_m\,Y_m^{(j)} - \mathbb{V}_m\,Y_m^{(j)}\,\Sigma_j) = \mathbb{V}_{m+1}\,Z_{m+1}\,\Psi_m^{(j)}, \\ &\Leftrightarrow R_0\,\Psi_0^{(j)} - (\mathbb{V}_{m+1}\,\widetilde{\mathbb{H}}_m\,Y_m^{(j)} - \mathbb{V}_m\,Y_m^{(j)}\,\Sigma_j) = \mathbb{V}_{m+1}\,Z_{m+1}\,\Psi_m^{(j)}. \end{aligned} \tag{22}$$

As $\mathbb{V}_{m+1}$ is orthonormal and $R_0 = V_1\,H_{1,0}$, we get:

$$\mathbb{V}_{m+1}\left(\widetilde{\mathbb{H}}_m\,Y_m^{(j)} - \begin{bmatrix} Y_m^{(j)} \\ 0_s \end{bmatrix}\Sigma_j + Z_{m+1}\,\Psi_m^{(j)}\right) = \mathbb{V}_{m+1}\,E_1^{(m+1)}\,H_{1,0}\,\Psi_0^{(j)},$$

which can be rewritten as

$$[\widetilde{\mathbb{H}}_m,\ Z_{m+1}]\begin{bmatrix} Y_m^{(j)} \\ \Psi_m^{(j)} \end{bmatrix} - \begin{bmatrix} Y_m^{(j)} \\ 0_s \end{bmatrix}\Sigma_j = E_1^{(m+1)}\,H_{1,0}\,\Psi_0^{(j)}.$$

Partitioning the $(m+1)s \times s$ matrix Z_{m+1} as $Z_{m+1} = \begin{bmatrix} Z_m \\ z_{m+1} \end{bmatrix}$, we obtain:

$$\begin{cases} [\mathbb{H}_m,\ Z_m]\begin{bmatrix} Y_m^{(j)} \\ \Psi_m^{(j)} \end{bmatrix} - Y_m^{(j)}\,\Sigma_j = E_1^{(m)}\,H_{1,0}\,\Psi_0^{(j)} \\ H_{m+1,m}\,(Y_m^{(j)})_m + z_{m+1}\,\Psi_m^{(j)} = 0 \end{cases},$$

where $(Y_m^{(j)})_m = {E_m^{(m)}}^T\,Y_m^{(j)}$. The last equality gives

$$\Psi_m^{(j)} = -z_{m+1}^{-1}\,H_{m+1,m}\,(Y_m^{(j)})_m, \tag{23}$$

and so

$$\mathbb{H}_m\,Y_m^{(j)} - Z_m\,z_{m+1}^{-1}\,H_{m+1,m}\,(Y_m^{(j)})_m - Y_m^{(j)}\,\Sigma_j = E_1^{(m)}\,H_{1,0}\,\Psi_0^{(j)}.$$

Denoting by

$$\mathbb{H}_m^Z = \mathbb{H}_m - Z_m\,z_{m+1}^{-1}\,H_{m+1,m}\,{E_m^{(m)}}^T, \tag{24}$$

which is also an upper block Hessenberg matrix, we see that $Y_m^{(j)}$ is the solution of the reduced Sylvester equation:

$$\mathbb{H}_m^Z\, Y_m^{(j)} - Y_m^{(j)}\, \Sigma_j = E_1^{(m)}\, H_{1,0}\, \Psi_0^{(j)}, \tag{25}$$

where $\Psi_m^{(j)}$ is given by (23). Finally and before summarizing the restarted shifted BGMRES algorithm, we point out that when restarting, the new initial residuals $\bar{R}_0$ for the base system and $\bar{R}_0^{(j)}$ for each add system are $\bar{R}_0 = R_m$ and $\bar{R}_0^{(j)} = R_m^{(j)}$ respectively. In this case, the collinearity condition on the new initial residual is still satisfied with $\bar{\Psi}_0^{(j)} = \Psi_m^{(j)}$.

Algorithm 5 Restarted shifted BGMRES with an unshifted base system.

Input: $A \in \mathbb{C}^{n\times n}$, $B = [b^{(1)}, \ldots, b^{(s)}] \in \mathbb{C}^{n\times s}$, $\{\sigma_j^{(i)}\}_{j=1,\ldots,k}^{i=1,\ldots,s} \subset \mathbb{C}$; $m \in \mathbb{N}$;

1: Set $X_0 = 0_{n\times s}$, $R_0 = B$ and $\mathcal{J} = \{1, \ldots, k\}$;
2: **for** each $j \in \mathcal{J}$ **do**
3: set: $\Sigma_j = \mathrm{diag}([\sigma_j^{(1)}, \ldots, \sigma_j^{(s)}])$, $X_0^{(j)} = X_0$; $\Psi_0^{(j)} = I_s$;
4: **end for**
5: Compute the reduced QR decomposition of R_0, i.e., $R_0 = V_1\, H_{1,0}$;
6: Generate $\mathbb{V}_{m+1}$, $\widetilde{\mathbb{H}}_m$ by Algorithm 1 applied to the pair (A, V_1);
7: Determine Y_m the solution of the least squares problem $\min_{Y\in\mathbb{C}^{ms\times s}} \|E_1^{(m+1)}\, H_{1,0} - \widetilde{\mathbb{H}}_m\, Y\|_F$;
8: Compute $Z_{m+1} = E_1^{(m+1)}\, H_{1,0} - \widetilde{\mathbb{H}}_m\, Y_m$; $R_m = \mathbb{V}_{m+1}\, Z_{m+1}$;
9: Compute $\mathbb{H}_m^Z = \mathbb{H}_m - Z_m\, z_{m+1}^{-1}\, H_{m+1,m}\, E_m^{(m)^T}$;
10: **for** each $j \in \mathcal{J}$ **do**
11: solve $\mathbb{H}_m^Z\, Y_m^{(j)} - Y_m^{(j)}\, \Sigma_j = E_1^{(m)}\, H_{1,0}\, \Psi_0^{(j)}$;
12: update $X_m^{(j)} \leftarrow X_m^{(j)} + \mathbb{V}_m\, Y_m^{(j)}$;
13: compute: $\Psi_m^{(j)} = -z_{m+1}^{-1}\, H_{m+1,m}\, Y_m^{(j)}$; $R_m^{(j)} = R_m\, \Psi_m^{(j)}$;
14: **end for**
15: Eliminate converged systems and update $\mathcal{J}$; If $\mathcal{J} = \emptyset$ exit;
16: **for** each $j \in \mathcal{J}$ **do**
17: update $X_0^{(j)} \leftarrow X_m^{(j)}$, $\Psi_0^{(j)} \leftarrow \Psi_m^{(j)}$; $R_0 = R_m$;
18: **end for**
19: goto line 5;

Remark 2 Before ending this subsection, we list below some remarks:

1. The residual norms of $R_m^{(j)}$ are available cheaply at each iteration without explicit computation of each correction $X_m^{(j)}$ since we have $\|R_m^{(j)}\|_F = \|R_m\, \Psi_m^{(j)}\|_F$. This avoids the need to carry out vector-matrix products with the large matrix A.
2. As was the case with the restarted shifted BFOM algorithm:

– The solutions $Y_m^{(j)}$ of the k Sylvester (25) can all be computed by solving the single Sylvester equation:

$$\mathbb{H}_m^Z \, \mathbb{Y}_m - \mathbb{Y}_m \, \Delta = \mathbb{E},$$

where the unknown $\mathbb{Y}_m = [Y_m^{(1)}, \ldots, Y_m^{(k)}] \in \mathbb{C}^{ms \times ks}$, the matrix $\Delta = \mathrm{diag}(\Sigma_1, \ldots, \Sigma_k) \in \mathbb{C}^{ks \times ks}$, and the right-hand side $\mathbb{E} = E_1^{(m)} \, H_{1,0} \, [\Psi_0^1, \ldots, \Psi_0^{(k)}] \in \mathbb{C}^{ms \times ks}$.

– The s columns $y_i^{(j)}$ of each block $Y_m^{(j)}$ $(j = 1, \ldots, k)$ can be obtained successively by solving the single linear systems:

$$(\mathbb{H}_m^Z - \sigma_i^{(j)} \, I_{ms}) \, y_i^{(j)} = E_1^{(m)} \, H_{1,0} \, (\Psi_0^{(j)})_i,$$

where $(\Psi_0^{(j)})_i$ is the i-th column of $\Psi_0^{(j)}$.

4.2 The shifted BGMRES algorithm with a variable shifted base system

Now, instead of the fixed base block system (2), we consider here a base block system that can change after restart. Specifically, in the hope of improving the convergence, this base block system will be selected as done in the previous section and so the index j_0 satisfies (12). As previously used, we take before restarting $X_0^{(j)} = 0_{n \times s}$ for $j = 1, \ldots, k$ and search for approximate solutions $X_m^{(j)} = X_0^{(j)} + \mathbb{V}_m \, Y_m^{(j)}$ where $\mathbb{V}_m$ is the orthonormal basis generated by the block Arnoldi process applied to the pair $(A, R_0^{(j_0)})$. Then, as in [38], applying the orthogonality condition:

$$R_m^{(j_0)} \perp \mathcal{L}_{j_0}(\mathbb{K}_m(A, R_0^{(j_0)})),$$

we easily check that $Y_m^{(j_0)}$ is the solution of the following least-squares problem

$$\min_{Y \in \mathbb{C}^{ms \times s}} \left\| E_1^{(m+1)} \, H_{1,0} - \left(\widetilde{\mathbb{H}}_m \, Y - \begin{bmatrix} Y \\ 0_s \end{bmatrix} \Sigma_{j_0} \right) \right\|_F, \tag{26}$$

and the corresponding residual is $R_m^{(j_0)} = \mathbb{V}_{m+1} \, Z_{m+1}^{(j_0)}$, where

$$Z_{m+1}^{(j_0)} = E_1^{(m+1)} \, H_{1,0} - \widetilde{\mathbb{H}}_m \, Y_m^{(j_0)} + \begin{bmatrix} Y_m^{(j_0)} \\ 0_s \end{bmatrix} \Sigma_{j_0}. \tag{27}$$

Notice that each $Y_m^{(j)}$ in the approximate solution $X_m^{(j)}$ of the add system (3) ($j = 1, \ldots, k$ and $j \neq j_0$) is determined by requiring the collinearity condition

$$R_m^{(j)} = R_m^{(j_0)} \, \Psi_m^{(j)}, \quad \text{where } \Psi_m^{(j)} \in \mathbb{C}^{s \times s}. \tag{28}$$

Based on this collinearity condition and as done in the case of the unshifted base system, we check that the correction $Y_m^{(j)}$ for $j \neq j_0$ satisfies:

$$[\widetilde{\mathbb{H}}_m,\ Z_{m+1}^{(j_0)}] \begin{bmatrix} Y_m^{(j)} \\ \Psi_m^{(j)} \end{bmatrix} - \begin{bmatrix} Y_m^{(j)} \\ 0_s \end{bmatrix} \Sigma_j = E_1^{(m+1)}\, H_{1,0}\, \Psi_0^{(j)}.$$

Partitioning the matrix $Z_{m+1}^{(j_0)}$ under the form $Z_{m+1}^{(j_0)} = \begin{bmatrix} Z_m^{(j_0)} \\ z_{m+1}^{(j_0)} \end{bmatrix}$, the previous equality is transformed as follows:

$$\begin{cases} \left[\mathbb{H}_m,\ Z_m^{(j_0)}\right] \begin{bmatrix} Y_m^{(j)} \\ \Psi_m^{(j)} \end{bmatrix} - Y_m^{(j)} \Sigma_j = E_1^{(m+1)}\, H_{1,0}\, \Psi_0^{(j)}, \\ H_{m+1,m}\, {E_m^{(m)}}^T\, Y_m^{(j)} + z_{m+1}^{(j_0)}\, \Psi_m^{(j)} = 0. \end{cases}$$

Then, $\Psi_m^{(j)}$ is obtained by

$$\Psi_m^{(j)} = -(z_{m+1}^{(j_0)})^{-1}\, H_{m+1,m}\, {E_m^{(m)}}^T\, Y_m^{(j)},$$

and $Y_m^{(j)}$ is the solution of the projected Sylvester equations

$$\mathbb{H}_m^Z\, Y_m^{(j)} - Y_m^{(j)}\, \Sigma_j = E_1^{(m)}\, H_{1,0}\, \Psi_0^{(j)},$$

in which $\mathbb{H}_m^Z = \mathbb{H}_m - Z_m^{(j_0)}\, (z_{m+1}^{(j_0)})^{-1}\, H_{m+1,m}\, {E_m^{(m)}}^T$.

Once all the approximate solutions have been computed and the converged block systems have been discarded, we proceed to the restart phase by selecting a new index $\bar{j}_0$ and taking $\bar{X}_0^{(j)} = X_m^{(j)}$, $\bar{R}_0^{(j)} = R_m^{(j)}$ and $\bar{\Psi}_0^{(j)} = (\Psi_m^{(\bar{j}_0)})^{-1}\, \Psi_m^{(j)}$. This ensures that the new starting residuals $\bar{R}_0^{(j)}$ are all collinear with $\bar{R}_0^{(\bar{j}_0)}$. The previous modifications are summarized in the following algorithm.

Algorithm 6 Restarted shifted BGMRES with a variable base system.

Input: $A \in \mathbb{C}^{n\times n}$, $B = [b^{(1)}, \ldots, b^{(s)}] \in \mathbb{C}^{n\times s}$, $\{\sigma_j^{(i)}\}_{j=1,\ldots,k}^{i=1,\ldots,s} \subset \mathbb{C}$; $m \in \mathbb{N}$;

1: Set $X_0 = 0_{n\times s}$, $\mathcal{J} = \{1, \ldots, k\}$ and $j_0 = 1$;
2: **for** each $j \in \mathcal{J}$ **do**
3: set: $\Sigma_j = \mathrm{diag}([\sigma_j^{(1)}, \ldots, \sigma_j^{(s)}])$, $X_0^{(j)} = X_0$; $R_0^{(j)} = B$; $\Psi_0^{(j)} = I_s$;
4: **end for**
5: Compute the reduced QR decomposition of $R_0^{(j_0)}$, i.e., $R_0^{(j_0)} = V_1 H_{1,0}$;
6: Generate $\mathbb{V}_{m+1}$, $\widetilde{\mathbb{H}}_m$ by Algorithm 1 applied to the pair (A, V_1);
7: Determine $Y_m^{(j_0)}$ the solution of the least squares problem $\min_{Y\in\mathbb{C}^{ms\times s}} \left\| E_1^{(m+1)} H_{1,0} - \left(\widetilde{\mathbb{H}}_m Y - \begin{bmatrix} Y \\ 0_s \end{bmatrix} \Sigma_{j_0}\right)\right\|_F$;
8: Compute $Z_{m+1}^{(j_0)} = E_1^{(m+1)} H_{1,0} - \widetilde{\mathbb{H}}_m Y_m^{(j_0)} + \begin{bmatrix} Y_m^{(j_0)} \\ 0_s \end{bmatrix} \Sigma_{j_0}$; $R_m^{(j_0)} = \mathbb{V}_{m+1} Z_{m+1}^{(j_0)}$;
9: Compute $\mathbb{H}_m^Z = \mathbb{H}_m - Z_m^{(j_0)} (z_{m+1}^{(j_0)})^{-1} H_{m+1,m} E_m^{(m)^T}$;
10: **for** $j \in \mathcal{J}$ **do**
11: **if** $j \neq j_0$ **then**
12: solve $\mathbb{H}_m^Z Y_m^{(j)} - Y_m^{(j)} \Sigma_j = E_1^{(m)^T} H_{1,0} \Psi_0^{(j)}$;
13: compute $\Psi_m^{(j)} = -z_{m+1}^{-1} H_{m+1,m} Y_m^{(j)}$; $R_m^{(j)} = R_m^{(j_0)} \Psi_m^{(j)}$;
14: **end if**
15: update $X_m^{(j)} \leftarrow X_m^{(j)} + \mathbb{V}_m Y_m^{(j)}$;
16: **end for**.
17: Eliminate converged systems and update $\mathcal{J}$; If $\mathcal{J} = \emptyset$ exit;
18: Select j_0 the index of the base block system using (12).
19: **for** each $j \in \mathcal{J}$ **do**
20: update $X_0^{(j)} \leftarrow X_m^{(j)}$, $\Psi_0^{(j)} \leftarrow (\Psi_m^{(j_0)})^{-1} \Psi_m^{(j)}$; $R_0^{(j)} \leftarrow R_m^{(j)}$;
21: **end for**.
22: goto line 5;

Remark 3 Since Σ_{j_0} is a diagonal matrix, the i-th column of $Y_m^{(j_0)} = [y_1^{(j_0)}, \ldots, y_s^{(j_0)}]$ the solution of the minimization problem (26) is

$$y_i^{(j_0)} = \arg\min_{y\in\mathbb{C}^{ms}} \| E_1^{(m+1)} (H_{1,0})_{:,i} - (\widetilde{\mathbb{H}}_m - \sigma_i^{(j_0)} I_{(m+1)s\times ms})\, y\|_2, \quad \text{for } i = 1, \ldots, s.$$

4.3 The deflated restarted shifted BGMRES method

In this subsection, taking inspiration from the deflation strategy proposed by Morgan [51], the restarted shifted BGMRES with deflation of eigenvalues (called rsh-BGMRES-D) for solving the matrix (3) is described in detail. In addition, we consider here the two cases:

- Case 1: rsh-BGMRES-D with a fixed base system.
- Case 2: rsh-BGMRES-D with a variable base system.

To begin, the definition of a harmonic Ritz pair is recalled since it plays an important role in the deflation procedure.

Definition 2 Assume that $\mathcal{U}$ is a generic subspace of $\mathbb{C}^n$. Let $C \in \mathbb{C}^{n\times n}$, $\theta \in \mathbb{C}$ and $y \in \mathcal{U}$. Then, the pair (θ, y) is a harmonic Ritz pair of C with respect to $\mathcal{U}$ if and only if it satisfies the following relation

$$C\, y - \theta\, y \perp C\, \text{range}(\mathcal{U}),$$

or equivalently, for the canonical scalar product,

$$\forall w \in C\, \text{range}(\mathcal{U}), \quad w^H\, (C\, y - \theta\, y) = 0.$$

The vector y is called a harmonic Ritz vector associated with the harmonic Ritz value θ.

Using similar arguments that were used for rsh-BFOM-D, we assume that $(\theta_i, \mathbb{V}_m\, g_i)$, for $i = 1, 2, \ldots, l$, are the harmonic Ritz pairs of A in range$(\mathbb{V}_m)$ and orthogonal to A range$(\mathbb{V}_m)$. Therefore, it follows that:

$$(A\, \mathbb{V}_m)^H\, (A\, \mathbb{V}_m\, g_i - \theta_i\, \mathbb{V}_m\, g_i) = 0, \quad \text{for } i = 1, 2, \ldots, l. \tag{29}$$

From (4) and since $\mathbb{V}_{m+1}^H\, \mathbb{V}_{m+1} = I_{(m+1)s}$, we get:

$$\widetilde{\mathbb{H}_m^H} \left(\widetilde{\mathbb{H}}_m\, g_i - \theta_i \begin{bmatrix} I_{ms} \\ 0_s \end{bmatrix} g_i \right) = 0, \tag{30}$$

which can be rewritten as the following generalized eigenvalue problem:

$$\left(\mathbb{H}_m^H\, \mathbb{H}_m + E_m^{(m)}\, H_{m+1,m}^H\, H_{m+1,m}\, E_m^{(m)^T} \right) g_i = \theta_i\, \mathbb{H}_m^H\, g_i,$$

or equivalently,

$$\left(\mathbb{H}_m + \mathbb{H}_m^{-H}\, E_m^{(m)}\, H_{m+1,m}^H\, H_{m+1,m}\, E_m^{(m)^T} \right) g_i = \theta_i\, g_i. \tag{31}$$

Therefore, the l targeted eigenpairs (θ_i, g_i) are obtained by solving the eigenvalue problem (31).

Before discussing the framework of rsh-BGMRES-D, the following lemma, whose proof is similar to that of Proposition 1 in [53], is presented. This lemma shows that the residuals of the matrix (2) and the residuals of the harmonic Ritz vectors belong to the subspace spanned by $Z_{m+1} = E_1^{(m+1)}\, H_{1,0} - \widetilde{\mathbb{H}}_m\, Y_m$.

Lemma 4 *Let $\mathbb{H}_m$ be full-rank and $Z_{m+1} = E_1^{(m+1)}\, H_{1,0} - \widetilde{\mathbb{H}}_m\, Y_m$. Then, there exist vectors $\alpha_i \in \mathbb{C}^s$, for $i = 1, 2, \ldots, l$, such that*

$$\widetilde{\mathbb{H}}_m\, g_i - \theta_i \begin{bmatrix} I_{ms} \\ 0_s \end{bmatrix} g_i = Z_{m+1}\, \alpha_i.$$

Proof It reveals from the relation (30) that

$$\widetilde{\mathbb{H}}_m\, g_i - \theta_i \begin{bmatrix} I_{ms} \\ 0_s \end{bmatrix} g_i \in \ker(\widetilde{\mathbb{H}_m^H}), \tag{32}$$

for $i = 1, 2, \ldots, l$. Since $\mathbb{H}_m$ is full-rank and

$$\dim(\mathrm{range}(\widetilde{\mathbb{H}_m^H})) + \dim(\ker(\widetilde{\mathbb{H}_m^H})) = ms + s,$$

it follows that $\dim(\ker(\widetilde{\mathbb{H}_m^H})) = s$. It is also well-known that

$$Y_m = \arg\min_{Y \in \mathbb{C}^{ms \times s}} \left\| E_1^{(m+1)}\, H_{1,0} - \widetilde{\mathbb{H}}_m\, Y \right\|_F,$$

then $\widetilde{\mathbb{H}}_m\, Y_m$ is the orthogonal projection of $E_1^{(m+1)}\, H_{1,0}$ onto $\mathrm{range}(\widetilde{\mathbb{H}}_m)$, and

$$E_1^{(m+1)}\, H_{1,0} - \widetilde{\mathbb{H}}_m\, Y_m \perp \mathrm{range}(\widetilde{\mathbb{H}}_m).$$

It implies that

$$E_1^{(m+1)}\, H_{1,0} - \widetilde{\mathbb{H}}_m\, Y_m \in \mathrm{range}(\widetilde{\mathbb{H}}_m)^{\perp} = \ker(\widetilde{\mathbb{H}_m^H}).$$

Therefore, since $\dim(\mathrm{range}(E_1^{(m+1)} H_{1,0} - \widetilde{\mathbb{H}}_m Y_m)) = s$, we have

$$\mathrm{range}(E_1^{(m+1)}\, H_{1,0} - \widetilde{\mathbb{H}}_m\, Y_m) = \ker(\widetilde{\mathbb{H}_m^H}),$$

and then, according to the relation (32), it is observed that all the vectors $\widetilde{\mathbb{H}}_m\, g_i - \theta_i \begin{bmatrix} I_{ms} \\ 0_s \end{bmatrix} g_i$, for $i = 1, \ldots, l$, belong to $\mathrm{range}(E_1^{(m+1)}\, H_{1,0} - \widetilde{\mathbb{H}}_m\, Y_m)$. This completes the proof. □

The next proposition has a key role in the implementation of the rsh-BGMRES-D method and characterizes a useful relation between the harmonic Ritz residual vectors $r_i^{hRitz} = A\, \mathbb{V}_m\, g_i - \theta_i\, \mathbb{V}_m\, g_i$, for $i = 1, 2, \ldots, l$, and the residual R_m associated to the approximate solution X_m.

Proposition 3 *Suppose that one cycle of Algorithm 5 applied to the matrix (3) is performed and let R_m be the residual associated to X_m an approximate solution of (2). Also, assume that $r_i^{hRitz} = A\, \mathbb{V}_m\, g_i - \theta_i\, \mathbb{V}_m\, g_i$, for $i = 1, 2, \ldots, l$, are the harmonic Ritz residual vectors. Then, there exists a matrix $T \in \mathbb{C}^{s \times l}$ such that*

$$\left[r_1^{hRitz}, \ldots, r_l^{hRitz} \right] = R_m\, T. \tag{33}$$

Proof Using relation (4), then for $i = 1, 2, \ldots, l$, we obtain:

$$\begin{aligned} r_i^{hRitz} &= A\, \mathbb{V}_m\, g_i - \theta_i\, \mathbb{V}_m\, g_i \\ &= \mathbb{V}_{m+1}\, \widetilde{\mathbb{H}}_m\, g_i - \theta_i\, \mathbb{V}_{m+1} \begin{bmatrix} I_{ms} \\ 0_s \end{bmatrix} g_i \\ &= \mathbb{V}_{m+1} \left(\widetilde{\mathbb{H}}_m\, g_i - \theta_i \begin{bmatrix} I_{ms} \\ 0_s \end{bmatrix} g_i \right). \end{aligned}$$

Moreover, the relation (29) yields that the residual R_m can be expressed as

$$R_m = \mathbb{V}_{m+1}\, Z_{m+1} = \mathbb{V}_{m+1}\, \left(E_1^{(m+1)}\, H_{1,0} - \widetilde{\mathbb{H}}_m\, Y_m \right).$$

On the other hand, Lemma 4 indicates that there exist vectors $t_i \in \mathbb{C}^s$ $(i = 1, \ldots, l)$ such that

$$\widetilde{\mathbb{H}}_m\, g_i - \theta_i \begin{bmatrix} I_{ms} \\ 0_s \end{bmatrix} g_i = \left(E_1^{(m+1)}\, H_{1,0} - \widetilde{\mathbb{H}}_m\, Y_m \right) t_i. \tag{34}$$

Pre-multiplying the above relation (34) by $\mathbb{V}_{m+1}$ gives

$$\mathbb{V}_{m+1} \left(\widetilde{\mathbb{H}}_m\, g_i - \theta_i \begin{bmatrix} I_{ms} \\ 0_s \end{bmatrix} g_i \right) = \mathbb{V}_{m+1} \left(E_1^{(m+1)}\, H_{1,0} - \widetilde{\mathbb{H}}_m\, Y_m \right) t_i.$$

Letting $T = [t_1, \ldots, t_l]$, it follows that

$$\left[r_1^{hRitz}, \ldots, r_l^{hRitz} \right] = R_m\, T,$$

which completes the proof. □

Now, after obtaining the l harmonic Ritz vectors and in order to update the next cycle, we use (33) to get:

$$\begin{aligned} A\, \mathbb{V}_m\, G_l &= \mathbb{V}_m\, G_l\, \mathrm{diag}(\theta_1, \ldots, \theta_l) + \mathbb{V}_{m+1}\, Z_{m+1}\, T \\ &= \mathbb{V}_{m+1} \begin{bmatrix} G_l\, \mathrm{diag}(\theta_1, \ldots, \theta_l) \\ 0_{s\times l} \end{bmatrix} + \mathbb{V}_{m+1}\, Z_{m+1}\, T \\ &= \mathbb{V}_{m+1} \left[\begin{bmatrix} G_l \\ 0_{s\times l} \end{bmatrix},\ Z_{m+1} \right] \begin{bmatrix} \mathrm{diag}(\theta_1, \ldots, \theta_l) \\ T \end{bmatrix}, \end{aligned}$$

where $G_l = [g_1, \ldots, g_l] \in \mathbb{C}^{ms\times l}$. Similarly, letting $G_l = Q_l\, \Gamma_l$ be the reduced QR factorization of G_l, we therefore get:

$$A\, \mathbb{V}_m\, Q_l\, \Gamma_l = \mathbb{V}_{m+1} \left[\begin{bmatrix} Q_l \\ 0_{s\times l} \end{bmatrix},\ Z_{m+1} \right] \begin{bmatrix} \Gamma_l\, \mathrm{diag}(\theta_1, \ldots, \theta_l) \\ T \end{bmatrix}.$$

Post-multiplying the above relation by Γ_l^{-1} implies that:

$$A\, \mathbb{V}_m\, Q_l = \mathbb{V}_{m+1} \underbrace{\left[\begin{bmatrix} Q_l \\ 0_{s\times l} \end{bmatrix},\ Z_{m+1} \right]}_{=:G_{l+1}} \begin{bmatrix} \Gamma_l\, \mathrm{diag}(\theta_1, \ldots, \theta_l) \\ T \end{bmatrix} \Gamma_l^{-1}. \tag{35}$$

Orthogonalizing the block matrix Z_{m+1} against $\begin{bmatrix} Q_l \\ 0_{s\times l} \end{bmatrix}$, we obtain the QR factorization $G_{l+1} = Q_{l+1}\, \Gamma_{l+1}$, where $Q_{l+1} \in \mathbb{C}^{(m+1)s\times(s+l)}$ and $\Gamma_{l+1} \in \mathbb{C}^{(s+l)\times(s+l)}$. Using this factorization, (35) can be expressed as:

$$A\, \mathbb{V}_m\, Q_l = \mathbb{V}_{m+1}\, Q_{l+1}\, \Gamma_{l+1} \begin{bmatrix} \Gamma_l\, \mathrm{diag}(\theta_1, \ldots, \theta_l) \\ T \end{bmatrix} \Gamma_l^{-1}.$$

Therefore, denoting by $\mathbb{V}_l^{new} = \mathbb{V}_m\, Q_l$, $\widetilde{\mathbb{H}}_l^{new} = \Gamma_{l+1} \begin{bmatrix} \Gamma_l\, \mathrm{diag}(\theta_1, \ldots, \theta_l) \\ T \end{bmatrix} \Gamma_l^{-1}$

and $\mathbb{V}_{l+s}^{new} = \mathbb{V}_{m+1}\, Q_{l+1}$, it can be readily verified that $\widetilde{\mathbb{H}}_l^{new} = Q_{l+1}^H\, \widetilde{\mathbb{H}}_m\, Q_l$ and we also have

$$A\, \mathbb{V}_l^{new} = \mathbb{V}_{l+s}^{new}\, \widetilde{\mathbb{H}}_l^{new},$$

which shows that the block Arnoldi relation holds.
Next, setting $\mathbb{V}_l = \mathbb{V}_l^{new}$, $V_{l+1} = \mathbb{V}_{l+s}^{new}(:, l+1 : l+s)$ and $\widetilde{\mathbb{H}}_l = \widetilde{\mathbb{H}}_l^{new}$, the block Arnoldi then carries out $(m - \frac{l}{s})$ iterations (in which it is assumed that l is divisible by s) with the starting block vector V_{l+1} to obtain at the end of the current cycle a block-Arnoldi relation similar to (4).

In the next subsections, we incorporate the deflation technique into the restarted shifted BGMRES with a fixed unshifted base system and into the restarted shifted BGMRES with a variable and shifted base system.

4.4 rsh-BGMRES-D with an unshifted base system

Now, we apply the first cycle of BGMRES to the block system (2) and generate the block Arnoldi relation: $A\, \mathbb{V}_m = \mathbb{V}_{m+1}\, \widetilde{\mathbb{H}}_m$. The residual R_m corresponding to the approximate solution $X_m = X_0 + \mathbb{V}_m\, Y_m \in \mathbb{C}^{n\times s}$ of the base system can be written as $R_m = \mathbb{V}_{m+1}\, (\mathbb{V}_{m+1}^H\, R_0 - \widetilde{\mathbb{H}}_m\, Y_m)$, where Y_m is the solution of the following least-squares problem:

$$\min_{Y\in\mathbb{C}^{ms\times s}} \left\| \mathbb{V}_{m+1}^H\, R_0 - \widetilde{\mathbb{H}}_m\, Y \right\|_F.$$

The approximate solutions of the add block systems can be updated by $X_m^{(j)} = X_0^{(j)} + \mathbb{V}_m\, Y_m^{(j)}$, where $Y_m^{(j)}$ is determined by the required collinearity condition (20). From (22), we have:

$$R_m^{(j)} = R_m\, \Psi_m^{(j)} \Longleftrightarrow \mathbb{V}_{m+1}\left(\widetilde{\mathbb{H}}_m\, Y_m^{(j)} - \begin{bmatrix} Y_m^{(j)} \\ 0_s \end{bmatrix} \Sigma_j + Z_{m+1}\, \Psi_m^{(j)}\right) = \mathbb{V}_{m+1}\, C_{m+1}^{(j)},$$

where $C_{m+1}^{(j)} = \mathbb{V}_{m+1}^H\, R_0\, \Psi_0^{(j)}$. Pre-multiplying both sides by $\mathbb{V}_{m+1}^H$ implies that

$$\widetilde{\mathbb{H}}_m\, Y_m^{(j)} - \begin{bmatrix} Y_m^{(j)} \\ 0_s \end{bmatrix} \Sigma_j + Z_{m+1}\, \Psi_m^{(j)} = C_{m+1}^{(j)}.$$

Thus, the sought parameters $Y_m^{(j)}$ and $\Psi_m^{(j)}$ can be obtained simultaneously by solving the following linear equations:

$$\left[\widetilde{\mathbb{H}}_m,\ Z_{m+1}\right] \begin{bmatrix} Y_m^{(j)} \\ \Psi_m^{(j)} \end{bmatrix} - \begin{bmatrix} Y_m^{(j)} \\ 0_s \end{bmatrix} \Sigma_j = C_{m+1}^{(j)}. \tag{36}$$

Partitioning the matrices Z_{m+1} and $C_{m+1}^{(j)}$ as $Z_{m+1} = \begin{bmatrix} Z_m \\ z_{m+1} \end{bmatrix}$ and $C_{m+1}^{(j)} = \begin{bmatrix} C_m^{(j)} \\ c_{m+1}^{(j)} \end{bmatrix}$, the relation (36) is written as follows:

$$\begin{cases} [\mathbb{H}_m,\ Z_m] \begin{bmatrix} Y_m^{(j)} \\ \Psi_m^{(j)} \end{bmatrix} - Y_m^{(j)}\, \Sigma_j = C_m^{(j)}, \\ H_{m+1,m}\, E_m^{(m)^T}\, Y_m^{(j)} + z_{m+1}\, \Psi_m^{(j)} = c_{m+1}^{(j)}. \end{cases}$$

From the second equation, we obtain:

$$\Psi_m^{(j)} = z_{m+1}^{-1} \left(c_{m+1}^{(j)} - H_{m+1,m}\, E_m^{(m)^T}\, Y_m^{(j)} \right),$$

and $Y_m^{(j)}$ is the solution of the following Sylvester matrix equation:

$$\mathbb{H}_m^Z\, Y_m^{(j)} - Y_m^{(j)}\, \Sigma_j = C_m^{(j)} - Z_m\, z_{m+1}^{-1}\, c_{m+1}^{(j)},$$

where $\mathbb{H}_m^Z = \mathbb{H}_m - Z_m\, z_{m+1}^{-1}\, H_{m+1,m}\, E_m^{(m)^T}$.
Notice that when restarting, the new initial residuals R_0 for the base system and $R_0^{(j)}$ for each add system are taken equal to R_m and $R_m^{(j)}$ respectively. Obviously, in this case, the new collinearity coefficients $\Psi_0^{(j)}$ for $j = 1, \ldots, k$ are $\Psi_0^{(j)} = \Psi_m^{(j)}$. The implementation of the rsh-BGMRES-D method with a fixed base system is given in Algorithm 7.

Algorithm 7 Restarted shifted block GMRES method with deflation of eigenvalues (rsh-BGMRES-D) with a fixed base system.

Input: $A \in \mathbb{C}^{n\times n}$, $B \in \mathbb{C}^{n\times s}$, $\left\{\sigma_j^{(i)}\right\}_{j=1,2,\ldots,k}^{i=1,2,\ldots,s}$ and $m \in \mathbb{N}$.

1: Set $X_0 = 0_{n\times s}$, $R_0 = B$ and $\mathcal{J} = \{1, 2, \ldots, k\}$;
2: **for** each $j \in \mathcal{J}$ **do**
3: set: $\Sigma_j = \text{diag}\left(\left[\sigma_j^{(1)}, \ldots, \sigma_j^{(s)}\right]\right)$, $X_0^{(j)} = X_0$; $\Psi_0^{(j)} = I_s$;
4: **end for**
5: Compute the reduced QR decomposition of R_0, i.e., $R_0 = V_1\, H_{1,0}$;
6: Generate $\mathbb{V}_{m+1}, \widetilde{\mathbb{H}}_m$ by Algorithm 1 applied to the pair (A, V_1);
7: Determine Y_m the solution of the least-squares problem $\min\limits_{Y\in\mathbb{C}^{ms\times s}} \left\| E_1^{(m+1)}\, H_{1,0} - \widetilde{\mathbb{H}}_m\, Y \right\|_F$;
8: Compute $Z_{m+1} = E_1^{(m+1)}\, H_{1,0} - \widetilde{\mathbb{H}}_m\, Y_m$; $R_m = \mathbb{V}_{m+1}\, Z_{m+1}$;
9: Compute $\mathbb{H}_m^Z = \mathbb{H}_m - Z_m\, z_{m+1}^{-1}\, H_{m+1,m}\, E_m^{(m)^T}$;
10: **for** each $j \in \mathcal{J}$ **do**
11: solve $\mathbb{H}_m^Z\, Y_m^{(j)} - Y_m^{(j)}\, \Sigma_j = E_1^{(m)}\, H_{1,0}\, \Psi_0^{(j)}$;
12: set $X_m^{(j)} = X_0^{(j)} + \mathbb{V}_m\, Y_m^{(j)}$; $\Psi_m^{(j)} = z_{m+1}^{-1}\, H_{m+1,m}\, E_m^{(m)^T}\, Y_m^{(j)}$; $R_m^{(j)} = R_m\, \Psi_m^{(j)}$;
13: update $X_0^{(j)} \leftarrow X_m^{(j)}$, $\Psi_0^{(j)} \leftarrow \Psi_m^{(j)}$ and $R_0 = R_m$;
14: **end for**
15: Compute the eigenpairs (θ_i, g_i) of the eigenvalue problem (31); Let $g_1, g_2, \ldots, g_l$ be the l eigenvectors corresponding to the l smallest eigenvalues in magnitude (If g_i is complex, consider the real and complex parts of g_i as two distinct eigenvectors. In this case, it may be necessary to adjust l); set $G_l = [g_1, g_2, \ldots, g_l] \in \mathbb{C}^{ms\times l}$.
16: Compute the reduced QR factorizations of G_l and G_{l+1}, i.e., $G_l = Q_l\, \Gamma_l$, $G_{l+1} = Q_{l+1}\, \Gamma_{l+1}$; % G_{l+1} is defined in (35)
17: Set $\mathbb{V}_l^{new} = \mathbb{V}_m\, Q_l$, $\mathbb{V}_{l+s}^{new} = \mathbb{V}_{m+1}\, Q_{l+1}$ and $\widetilde{\mathbb{H}}_l^{new} = Q_{l+1}^H\, \widetilde{\mathbb{H}}_m\, Q_l$;
18: Set $\mathbb{V}_l = \mathbb{V}_l^{new}$, $\widetilde{\mathbb{H}}_l = \widetilde{\mathbb{H}}_l^{new}$, $\mathbb{V}_{l+s} = \mathbb{V}_{l+s}^{new}$ and apply $m - \frac{l}{s}$ steps of the block Arnoldi process to extend $\mathbb{V}_{l+s}$ and $\widetilde{\mathbb{H}}_l$ to $\mathbb{V}_{m+1}$ and $\widetilde{\mathbb{H}}_m$ respectively;
19: Determine Y_m the solution of the least-squares problem $\min\limits_{Y\in\mathbb{C}^{ms\times s}} \left\| \mathbb{V}_{m+1}^H\, R_0 - \widetilde{\mathbb{H}}_m\, Y \right\|_F$;
20: Compute $Z_{m+1} = \mathbb{V}_{m+1}^H\, R_0 - \widetilde{\mathbb{H}}_m\, Y_m$, $R_m = \mathbb{V}_{m+1}\, Z_{m+1}$; $\mathbb{H}_m^Z = \mathbb{H}_m - Z_m\, z_{m+1}^{-1}\, H_{m+1,m}\, E_m^{(m)^T}$;
21: **for** each $j \in \mathcal{J}$ **do**
22: compute $C_{m+1}^{(j)} = \mathbb{V}_{m+1}^H\, R_0\, \Psi_0^{(j)}$; % $C_{m+1}^{(j)^T} = [C_m^{(j)^T},\ c_{m+1}^{(j)^T}]$;
23: solve $\mathbb{H}_m^Z\, Y_m^{(j)} - Y_m^{(j)}\, \Sigma_j = C_m^{(j)} - Z_m\, z_{m+1}^{-1}\, c_{m+1}^{(j)}$;
24: set $X_m^{(j)} = X_0^{(j)} + \mathbb{V}_m\, Y_m^{(j)}$, $\Psi_m^{(j)} = z_{m+1}^{-1}\left(c_{m+1}^{(j)} - H_{m+1,m}\, E_m^{(m)^T}\, Y_m^{(j)}\right)$ and $R_m^{(j)} = R_m\, \Psi_m^{(j)}$;
25: **end for**
26: Eliminate converged systems; Update $\mathcal{J}$, if $\mathcal{J} = \emptyset$ exist.
27: **for** each $j \in \mathcal{J}$ **do**
28: update $X_0^{(j)} \leftarrow X_m^{(j)}$, $\Psi_0^{(j)} \leftarrow \Psi_m^{(j)}$, and $R_0 = R_m$;
29: **end for**
30: go to line 15;

4.5 rsh-BGMRES-D with a variable shifted base system

Suppose that the first cycle has been performed and consider the block Arnoldi relation $A\mathbb{V}_m = \mathbb{V}_{m+1}\widetilde{\mathbb{H}}_m$. Then, letting j_0 satisfy (12) as the index of the selected base block system, we set $R_0^{(j_0)} = R_m^{(j_0)}$ and update $\Psi_0^{(j)} = (\Psi_m^{(j_0)})^{-1}\,\Psi_m^{(j)}$. In a similar way to the case of the rsh-BGMRES-D with a fixed base system, we adapt the deflation procedure to construct the new block Arnoldi relation. To achieve this goal, it is assumed that the new block Arnoldi relation is

$$A\,\mathbb{V}_m^{new} = \mathbb{V}_{m+1}^{new}\,\widetilde{\mathbb{H}}_m^{new}.$$

Setting $\mathbb{V}_m = \mathbb{V}_m^{new}$, $\mathbb{V}_{m+1} = \mathbb{V}_{m+1}^{new}$ and $\widetilde{\mathbb{H}}_m = \widetilde{\mathbb{H}}_m^{new}$ yield $A\mathbb{V}_m = \mathbb{V}_{m+1}\widetilde{\mathbb{H}}_m$. Now, the approximate solution $X_m^{(j_0)}$ and the corresponding residual $R_m^{(j_0)}$ of the base system can be updated easily. In fact, we have:

$$X_m^{(j_0)} = X_0^{(j_0)} + \mathbb{V}_m\,Y_m^{(j_0)} \quad \text{and} \quad R_m^{(j_0)} = \mathbb{V}_{m+1}\,Z_{m+1}^{(j_0)},$$

where

$$Z_{m+1}^{(j_0)} = \mathbb{V}_{m+1}^H\,R_0^{(j_0)} - \widetilde{\mathbb{H}}_m\,Y_m^{(j_0)} + \begin{bmatrix} Y_m^{(j_0)} \\ 0_s \end{bmatrix}\Sigma_{j_0},$$

and

$$Y_m^{(j_0)} = \underset{Y\in\mathbb{C}^{ms\times s}}{\arg\min}\left\|\mathbb{V}_{m+1}^H\,R_0^{(j_0)} - \left(\widetilde{\mathbb{H}}_m\,Y - \begin{bmatrix} Y \\ 0_s \end{bmatrix}\Sigma_{j_0}\right)\right\|_F.$$

For the add systems, the approximate solutions $X_m^{(j)}$ can be updated as $X_m^{(j)} = X_0^{(j)} + \mathbb{V}_m\,Y_m^{(j)}$ where the corrections $Y_m^{(j)}$ are obtained by using the collinearity condition (28). Indeed, we have:

$$R_m^{(j)} = R_m^{(j_0)}\,\Psi_m^{(j)} \Longleftrightarrow R_0^{(j)} - \left(A\,\mathbb{V}_m\,Y_m^{(j)} - \mathbb{V}_m\,Y_m^{(j)}\Sigma_j\right) = \mathbb{V}_{m+1}\,Z_{m+1}^{(j_0)}\,\Psi_m^{(j)}.$$

Analogously to the case of an unshifted base system, we let $C_{m+1}^{(j)} = \mathbb{V}_{m+1}^H\,R_0^{(j_0)}\,\Psi_0^{(j)}$ and we partition $Z_{m+1}^{(j_0)}$ and $C_{m+1}^{(j)}$ as $Z_{m+1}^{(j_0)} = \begin{bmatrix} Z_m^{(j_0)} \\ z_{m+1}^{(j_0)} \end{bmatrix}$ and $C_{m+1}^{(j)} = \begin{bmatrix} C_m^{(j)} \\ c_{m+1}^{(j)} \end{bmatrix}$ respectively. Then, it follows that:

$$\Psi_m^{(j)} = (z_{m+1}^{(j_0)})^{-1}\left(c_{m+1}^{(j)} - H_{m+1,m}\,E_m^{(m)^T}\,Y_m^{(j)}\right),$$

and $Y_m^{(j)}$ is the solution the reduced Sylvester equation:

$$\mathbb{H}_m^Z\,Y_m^{(j)} - Y_m^{(j)}\,\Sigma_j = C_m^{(j)} - Z_m^{(j_0)}\,(z_{m+1}^{(j_0)})^{-1}\,c_{m+1}^{(j)},$$

where $\mathbb{H}_m^z = \widetilde{\mathbb{H}}_m - Z_m^{(j_0)}\,(z_{m+1}^{(j_0)})^{-1}\,H_{m+1,m}\,E_m^{(m)^T}$.

Once all the approximate solutions have been computed and the converged block systems have been discarded, we proceed to the restart phase by selecting a new index $\bar{j}_0$ and taking $\bar{X}_0^{(j)} = X_m^{(j)}$, $\bar{\Psi}_0^{(j)} = (\Psi_m^{(\bar{j}_0)})^{-1}\,\Psi_m^{(j)}$ and $\bar{R}_0^{(j)} = R_m^{(j)}$. This ensure that the new starting residuals $\bar{R}_0^{(j)}$ are all collinear with $\bar{R}_0^{(\bar{j}_0)}$. The discussion, presented in this section, is summarized in Algorithm 8.

Algorithm 8 Restarted shifted block GMRES method with deflation of eigenvalues (rsh-BGMRES-D) with a variable base system.

Input: $A \in \mathbb{C}^{n\times n}$, $B \in \mathbb{C}^{n\times s}$, $\left\{\sigma_j^{(i)}\right\}_{j=1,2,\ldots,k}^{i=1,2,\ldots,s}$ and $m \in \mathbb{N}$.

1: Set $\mathcal{J} = \{1, 2, \ldots, k\}$ and $j_0 = 1$;
2: **for** each $j \in \mathcal{J}$ **do**
3: set: $\Sigma_j = \text{diag}\left(\left[\sigma_j^{(1)}, \ldots, \sigma_j^{(s)}\right]\right)$, $X_0^{(j)} = 0_{n\times s}$, $R_0^{(j)} = B$ and $\Psi_0^{(j)} = I_s$;
4: **end for**
5: Compute the reduced QR decomposition of $R_0^{(j_0)}$, i.e., $R_0^{(j_0)} = V_1\, H_{1,0}$;
6: Generate $\mathbb{V}_{m+1}$, $\widetilde{\mathbb{H}}_m$ by Algorithm 1 applied to the pair (A, V_1);
7: Determine $Y_m^{(j_0)}$ the solution of the least squares problem $\min\limits_{Y\in\mathbb{C}^{ms\times s}} \left\| E_1^{(m+1)}\, H_{1,0} - \widetilde{\mathbb{H}}_m\, Y + \begin{bmatrix} Y \\ 0_s \end{bmatrix} \Sigma_{j_0} \right\|_F$;
8: Compute $Z_{m+1}^{(j_0)} = E_1^{(m+1)}\, H_{1,0} - \widetilde{\mathbb{H}}_m\, Y_m^{(j_0)} + \begin{bmatrix} Y_m^{(j_0)} \\ 0_s \end{bmatrix} \Sigma_{j_0}$ and $R_m^{(j_0)} = \mathbb{V}_{m+1}\, Z_{m+1}^{(j_0)}$;
9: Compute $\mathbb{H}_m^Z = \mathbb{H}_m - Z_m^{(j_0)}\, (z_{m+1}^{(j_0)})^{-1}\, H_{m+1,m}\, {E_m^{(m)}}^T$;
10: **for** each $j \in \mathcal{J}$ **do**
11: **if** $j \neq j_0$ **then**
12: solve $\mathbb{H}_m^Z\, Y_m^{(j)} - Y_m^{(j)}\, \Sigma_j = E_1^{(m)}\, H_{1,0}\, \Psi_0^{(j)}$;
13: **end if**
14: set $X_m^{(j)} = X_0^{(j)} + \mathbb{V}_m\, Y_m^{(j)}$, $\Psi_m^{(j)} = -(z_{m+1}^{(j_0)})^{-1}\, H_{m+1,m}\, {E_m^{(m)}}^T\, Y_m^{(j)}$ and $R_m^{(j)} = R_m^{(j_0)} \Psi_m^{(j)}$;
15: update $X_0^{(j)} \leftarrow X_m^{(j)}$, $\Psi_0^{(j)} \leftarrow \Psi_m^{(j)}$, and $R_0^{(j)} = R_m^{(j)}$;
16: **end for**
17: Compute the eigenpairs (θ_i, g_i) of the eigenvalue problem (31); Let $g_1, g_2, \ldots, g_l$ be l the eigenvectors corresponding to the l smallest eigenvalues in magnitude (If g_i is complex consider the real and complex parts of g_i as two distinct eigenvectors. In this case, it may be necessary to adjust l); set $G_l = [g_1, g_2, \ldots, g_l] \in \mathbb{C}^{ms\times l}$.
18: Compute the reduced QR factorizations of G_l and G_{l+1}, i.e., $G_l = Q_l\, \Gamma_l$, $G_{l+1} = Q_{l+1}\, \Gamma_{l+1}$; % G_{l+1} is defined in (35)
19: Set $\mathbb{V}_l^{new} = \mathbb{V}_m\, Q_l$, $\mathbb{V}_{l+s}^{new} = \mathbb{V}_{m+1}\, Q_{l+1}$ and $\widetilde{\mathbb{H}}_l^{new} = Q_{l+1}^H\, \widetilde{\mathbb{H}}_m\, Q_l$;
20: Set $\mathbb{V}_l = \mathbb{V}_l^{new}$, $\widetilde{\mathbb{H}}_l = \widetilde{\mathbb{H}}_l^{new}$, $\mathbb{V}_{l+s} = \mathbb{V}_{l+s}^{new}$ and apply $m - \frac{l}{s}$ steps of the block Arnoldi process to extend $\mathbb{V}_{l+s}$ and $\widetilde{\mathbb{H}}_l$ to $\mathbb{V}_{m+1}$ and $\widetilde{\mathbb{H}}_m$ respectively;
21: Determine $Y_m^{(j_0)}$ the solution of the least squares problem (26);
22: Compute $Z_{m+1}^{(j_0)}$ using (27) and set $R_m^{(j_0)} = \mathbb{V}_{m+1}\, Z_{m+1}^{(j_0)}$; $\mathbb{H}_m^Z = \mathbb{H}_m - Z_m^{(j_0)}\, (z_{m+1}^{(j_0)})^{-1}\, H_{m+1,m}\, {E_m^{(m)}}^T$;
23: **for** each $j \in \mathcal{J}$ **do**
24: compute $C_{m+1}^{(j)} = \mathbb{V}_{m+1}^H\, R_0^{(j_0)}\, \Psi_0^{(j)}$; % ${C_{m+1}^{(j)}}^T = [{C_m^{(j)}}^T,\ {c_{m+1}^{(j)}}^T]$;
25: **if** $j \neq j_0$ **then**
26: solve $\mathbb{H}_m^Z\, Y_m^{(j)} - Y_m^{(j)}\, \Sigma_j = C_m^{(j)} - Z_m^{(j_0)}\, (z_{m+1}^{(j_0)})^{-1}\, c_{m+1}^{(j)}$;
27: **end if**
28: compute $X_m^{(j)} = X_0^{(j)} + \mathbb{V}_m\, Y_m^{(j)}$; $\Psi_m^{(j)} = (z_{m+1}^{(j_0)})^{-1} \left(c_{m+1}^{(j)} - H_{m+1,m}\, {E_m^{(m)}}^T\, Y_m^{(j)}\right)$ and $R_m^{(j)} = R_m^{(j_0)}\, \Psi_m^{(j)}$;
29: **end for**
30: Eliminate converged systems; Update $\mathcal{J}$, if $\mathcal{J} = \emptyset$ exist.
31: Select j_0 the index of the base block system using (12).
32: **for** each $j \in \mathcal{J}$ **do**
33: update $X_0^{(j)} \leftarrow X_m^{(j)}$; $\Psi_0^{(j)} \leftarrow (\Psi_m^{(j_0)})^{-1}\, \Psi_m^{(j)}$; $R_0^{(j)} = R_m^{(j)}$;
34: **end for**
35: go to line 5;

5 Numerical experiments

In this section, we give some experimental results to compare the different proposed methods and show their effectiveness. In what follows, "the restarted shifted BGMRES algorithm with an unshifted base system" and "the shifted BGMRES algorithm with a variable shifted base system" are denoted by "rsh-BGMRES-v1" and "rsh-BGMRES-v2," respectively.

In the first part our experiments, we use some test matrices from the University of Florida Matrix Collection [63]. In addition, two test matrices "**bidiag**" and "**tridiag**," which have been introduced in [51, 64], are applied. The **bidiag** matrix is bidiagonal in which the diagonal elements are $-2, -1, 1, \ldots, 998$, and the super diagonal elements are all equal to 0.1. The **tridiag** matrix is tridiagonal in which the diagonal elements are $0.1, 0.2, 0.3, 0.4, 0.5, 6, \ldots, 1000$, and the sub and the super diagonal elements are all equal to 1. The details of these test matrices are provided in Table 1. In the next part, we consider some matrices from the quantum chromodynamics application.

In all the experiments—except otherwise stated—the $s \times s$ diagonal matrix Σ_j, for $j = 1, 2, \ldots, k$, is randomly generated with a uniform distribution on $[-2, 0]$. The initial guess is considered $X_0 = 0_{n\times s}$. The columns of the right-hand side B were generated randomly and their coefficients were uniformly distributed in $[0, 1]$. For all the different methods, a maximum of 501 restarts was allowed and the algorithms were stopped as soon as the residual norm $\|R_m^{(j)}\|_F \leq \epsilon \|R_0^{(j)}\|_F$ with $\epsilon = 10^{-10}$. Note that the notation "-" in a table of results refers to the case where a method does not converge within 501 restarts. All the experiments were performed on a computer with an Intel Core i5 processor at 2.00 GHz and 4GB of RAM. The algorithms were coded in Matlab Release 2015a. The machine precision was equal to $2.22\,10^{-16}$.

Table 1 Information of the test matrices used in examples 1 to 4

Name	Size	Nonzeros	Type	Application area
nos5	468	5172	Real symmetric	Structural problem
sherman4	1104	3786	Real unsymmetric	Computational fluid dynamics problem
raefsky1	3242	293,409	Real unsymmetric	Computational fluid dynamics problem
poisson3Da	13,514	352,762	Real unsymmetric	Computational fluid dynamics problem
memplus	17,758	99,147	Real unsymmetric	Circuit simulation problem
wang3	26,064	177,168	Real unsymmetric	Semiconductor device problem
wathen100	30,401	471,601	Real symmetric	Random 2D/3D problem
wathen120	36,441	565,761	Real symmetric	Random 2D/3D problem
Dubcova2	65,025	1,030,225	Real symmetric	2D/3D problem
poisson3Db	85,623	2,374,949	Real unsymmetric	Computational fluid dynamics problem
bidiag	1000	1999	Real unsymmetric	Academic
tridiag	1000	2998	Real symmetric	Academic

In the following numerical results, the parameters m and l denote the dimension of the search subspace and the number of Ritz vectors (or harmonic Ritz vectors) added to the search subspace, respectively. Furthermore, for the sake of comparison, the number of matrix-vector products (denoted by mv), the Frobenius norm of the maximum true residual (denoted by mnres), and the total CPU time in terms of seconds (denoted by CPU) are illustrated in tables related to the examples.

Example 1 In this example, we assess the performance of the rsh-BFOM, rsh-BGMRES-v1, and rsh-BGMRES-v2 methods and examine the applicability of their deflated versions. The test matrices for this example are **nos5**, **sherman4**, **memplus**, **wang3**, **wathen120**, **bidiag**, and **tridiag**. Here, the dimension of the search subspace m is taken from $\{20, 30\}$, and the number of harmonic Ritz values is assumed to be equal to 10, i.e., $l = 10$. We also set $s = 5$ and $k = 3$.

Table 2 reports the number of matrix-vector products, maximum true residual norm, and CPU time in terms of the different values of m.

It can be seen from this table that the rsh-BGMRES-v2 method outperforms the rsh-BFOM and rsh-BGMRES-v1 methods in terms of the number of matrix-vector products or CPU time in many cases. In particular, for the cases of "**bidiag** when $m = 20, 30$" and "**tridiag** when $m = 30$," the rsh-BFOM and rsh-BGMRES-v1 methods fail to converge within 501 restarts, while the rsh-BGMRES-v2 method enjoys a high level of performance.

From Table 2, it is also observed that the deflated version of the rsh-BFOM and rsh-BGMRES-v1 methods needs less matrix-vector products and CPU time than the non-deflated ones. For instance, in the cases of "**bidiag** when $m = 20, 30$" and "**tridiag** when $m = 20, 30$," the rsh-BFOM and rsh-BGMRES-v1 methods do not converge within 501 restarts, while the deflated version of these methods performs perfectly well. In addition, the rsh-BGMRES-D-v1 method converges in less matrix-vector products and CPU time than the other comparison methods. This table also demonstrates that, except for the test matrices **nos5** and **sherman4** when $m = 20, 30$, the rsh-BGMRES-D-v2 method requires less matrix-vector products and CPU time than the rsh-BGMRES-v2 method. It is also worthwhile to note that in cases such as **nos5** or **sherman4**, the CPU time corresponds to the convergence of rsh-BGMRES-D-v2 is greater than that of rsh-BGMRES-v2 even though the first method requires less matrix-vector products than the other one. The reason for this situation lies in the fact that in case the harmonic Ritz value information is applied, it is needed to solve a generalized eigenvalue problem and to sort the eigenvalues. In conclusion, we may infer from the above observations that the use of the eigenvalue deflation technique in the rsh-BFOM, rsh-BGMRES-v1, and rsh-BGMRES-v2 methods improves the efficiency of the original methods.

Example 2 In this example, the test matrices are **nos5**, **sherman4**, **wathen120**, **Dubcova2**, **bidiag**, and **tridiag**. The value of s is taken from $\{3, 5\}$, the value of k is fixed as 3, and the values of the parameters m and l are set as 20 and 15, respectively.

Table 3 displays the number of matrix-vector products, maximum true residual norm, and CPU time in terms of the different values of s.

Table 2 Numerical results for example 1 with $s = 5$, $m = 20, 30$, $k = 3$, and $l = 10$

		$m = 20$			$m = 30$		
Test problem	Method	mv	mnres	CPU	mv	mnres	CPU
nos5	rsh-BFOM	22,600	2.8107e−09	8.7969	13,950	2.5515e−09	6.2656
	rsh-BFOM-D	3160	1.9502e−09	2.4688	2390	6.9215e−10	2.6719
	rsh-BGMRES-v1	19,000	2.8186e−08	9.4063	12150	2.6601e−09	7.8750
	rsh-BGMRES-D-v1	2890	1.8278e−09	2.3281	2110	1.7984e−09	2.6250
	rsh-BGMRES-v2	12,400	2.6855e−09	6.6875	9150	2.4940e−09	6.6094
	rsh-BGMRES-D-v2	6080	3.9500e−09	6.8281	4930	6.0492e−09	7.4688
sherman4	rsh-BFOM	1200	6.6441e−10	0.7968	900	2.1881e−09	0.8281
	rsh-BFOM-D	550	3.2294e−10	0.6093	430	2.6012e−09	0.6250
	rsh-BGMRES-v1	1300	1.9355e−09	1.0469	1050	2.4728e−09	1.0625
	rsh-BGMRES-D-v1	460	1.3464e−09	0.5781	430	3.8605e−09	0.6175
	rsh-BGMRES-v2	1000	3.0184e−09	0.7812	750	2.3152e−09	0.8750
	rsh-BGMRES-D-v2	950	1.4488e−08	1.3281	580	4.0279e−05	1.1563
memplus	rsh-BFOM	200	7.9429e−09	1.8750	300	3.9373e−10	3.7656
	rsh-BFOM-D	190	7.9429e−09	1.7344	290	3.9373e−10	3.2969
	rsh-BGMRES-v1	200	1.4348e−09	1.8906	300	5.1926e−10	3.7188
	rsh-BGMRES-D-v1	190	1.7599e−12	1.7206	290	6.2127e−10	3.1094
	rsh-BGMRES-v2	200	7.6338e−09	1.9844	300	5.2733e−09	3.7813
	rsh-BGMRES-D-v2	190	2.5941e−06	1.9063	290	2.2206e−04	3.5781
wang3	rsh-BFOM	200	1.0611e−09	3.0938	300	8.0289e−13	5.8906
	rsh-BFOM-D	190	1.0611e−09	2.7656	290	8.0289e−13	5.5000
	rsh-BGMRES-v1	200	1.1773e−12	3.1563	300	7.5715e−13	5.9688
	rsh-BGMRES-D-v1	190	6.4842e−09	2.0625	290	1.0018e−09	5.4031
	rsh-BGMRES-v2	200	1.3365e−12	3.0781	300	7.4894e−13	6.0156
	rsh-BGMRES-D-v2	190	8.3149e−06	3.0156	290	3.5338e−10	5.8906
wathen120	rsh-BFOM	6000	2.0014e−08	167.52	2850	1.8597e−08	109.09
	rsh-BFOM-D	1810	2.3124e−08	63.703	1410	1.2208e−08	62.672
	rsh-BGMRES-v1	5500	2.2777e−08	160.50	3000	1.4634e−08	119.23
	rsh-BGMRES-D-v1	1720	1.7338e−08	56.672	1410	1.5710e−08	60.484
	rsh-BGMRES-v2	4900	3.4170e−08	142.83	2850	1.1646e−08	113.28
	rsh-BGMRES-D-v2	2850	4.6137e−08	91.297	2320	3.4796e−08	97.781

From this table, it turns out that

- For the cases of "**sherman4** and **wathen120** when $s = 3, 5$" and "**Dubcova2** when $s = 5$," the rsh-BFOM method needs less matrix-vector products or CPU time than rsh-BGMRES-v1 and rsh-BGMRES-v2.
- For some cases, such as "**nos5** when $s = 3, 5$," "**Dubcova2** when $s = 3$," "**bidiag** when $s = 3, 5$," and "**tridiag** when $s = 5$," the rsh-BGMRES-v2 method is superior to the rsh-BFOM and rsh-BGMRES-v1 methods.

Table 2 (continued)

Test problem	Method	$m = 20$			$m = 30$		
		mv	mnres	CPU	mv	mnres	CPU
bidiag	rsh-BFOM	-	-	-	-	-	-
	rsh-BFOM-D	820	5.2023e−10	2.4375	850	8.5163e−10	3.1875
	rsh-BGMRES-v1	-	-	-	-	-	-
	rsh-BGMRES-D-v1	820	2.7902e−09	2.2031	850	3.5940e−10	3.0625
	rsh-BGMRES-v2	4600	1.4413e−04	10.234	2700	2.2761e−09	6.9844
	rsh-BGMRES-D-v2	2850	4.2540e−08	7.8906	1740	1.4768e−07	6.2656
tridiag	rsh-BFOM	-	-	-	-	-	-
	rsh-BFOM-D	820	2.8806e−09	2.5625	710	3.5091e−10	2.3125
	rsh-BGMRES-v1	-	-	-	-	-	-
	rsh-BGMRES-D-v1	820	1.4243e−09	2.0938	710	1.6357e−10	2.2188
	rsh-BGMRES-v2	-	-	-	3000	2.5907e−09	7.7188
	rsh-BGMRES-D-v2	2660	1.0421e−08	7.3125	1740	8.0097e−07	5.9844

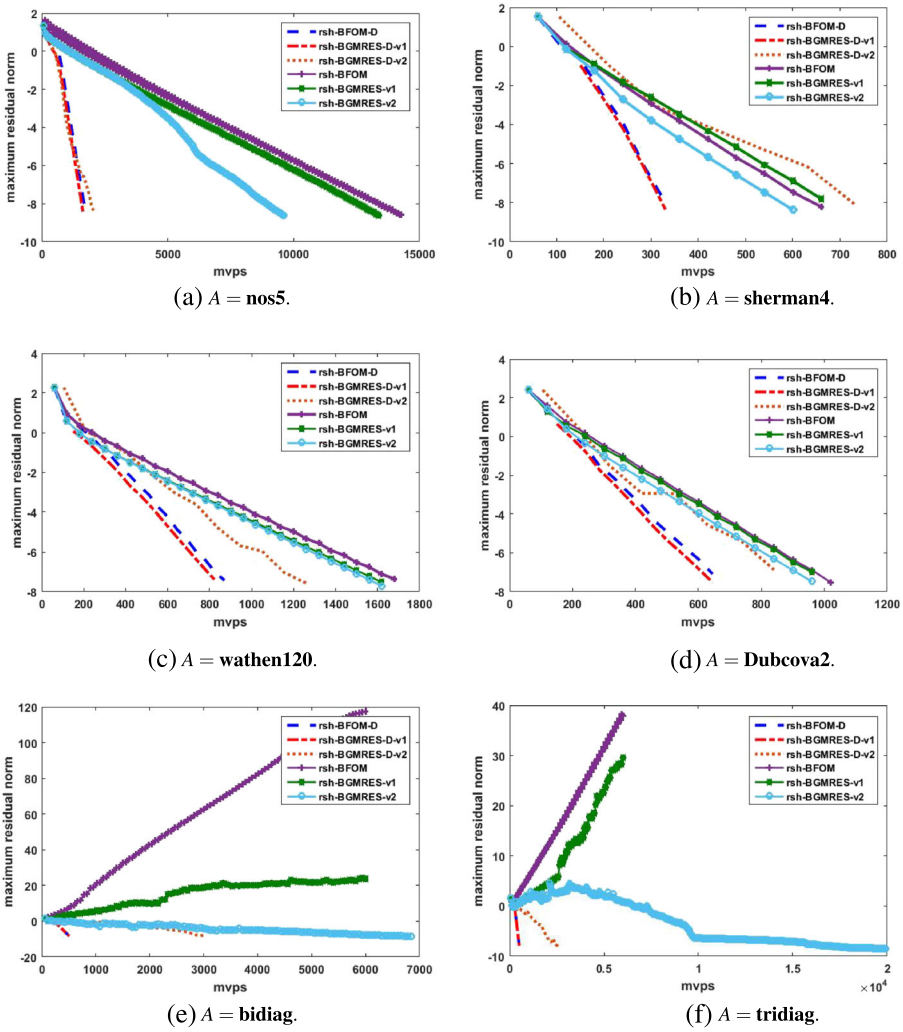
As indicated in Table 3, except for the case of "**Dubcova2** when $s = 5$," the required matrix-vector products and CPU time for the rsh-BFOM-D method are less than those of rsh-BFOM. Moreover, except for the cases of "**sherman4** when $s = 3, 5$," and "**wathen120** and **Dubcova2** when $s = 5$," we can see that the rsh-BGMRES-D-v2 method works better than the rsh-BGMRES-v2 method in terms of the number of matrix-vector products or CPU time. On the other hand, similar to example 1, the reported numerical results disclose the superiority of the rsh-BGMRES-D-v1 method over the other comparison methods.

In order to make a fair comparison, Fig. 1 depicts the $\log_{10}$ plot of the maximum residual norms with respect to the number of matrix-vector products for the proposed methods when $s = 3$. We observe that the rsh-BFOM-D and rsh-BGMRES-D-v1 methods not only converge more rapidly than the other methods, but they also achieve more accurate residual norms. In fact, for the test matrices **bidiag** and **tridiag**, it can be seen that rsh-BFOM and rsh-BGMRES-v1 begin to diverge significantly, while the rsh-BFOM-D and rsh-BGMRES-D-v1 methods converge very fast. It should also be noted that, for **wathen120** and **Dubcova2**, the rsh-BGMRES-D-v1 method slightly outperforms rsh-BFOM-D. Finally, a comparison among the proposed deflated methods reveals that the performance of rsh-BGMRES-D-v2 is not satisfying as well as that of rsh-BFOM-D and rsh-BGMRES-D-v1.

Example 3 In this example, the different values of l are considered in order to further evaluate the efficiency of the deflated version of the rsh-BFOM, rsh-BGMRES-v1, and rsh-BGMRES-v2 methods. In addition to this, we compare the effectiveness of our proposed deflated methods to that of GMRES-DR with shifts (GMRES-DR-Sh) proposed by Darnell et al. [33]. In the following, we use the five test matrices **nos5**, **memplus**, **sherman4**, **wathen120**, and **Dubcova2**. The value of the parameter l is

Table 3 Numerical results for example 2 with $s = 3, 5$, $m = 20$, $k = 3$, and $l = 15$

$m = 20$		$s = 3$			$s = 5$		
Test problem	Method	mv	mnres	CPU	mv	mnres	CPU
nos5	rsh-BFOM	14,280	2.0949e−09	4.7813	23,300	2.6213e−09	8.0781
	rsh-BFOM-D	1725	1.8782e−09	1.3594	2310	6.4439e−10	2.0156
	rsh-BGMRES-v1	13,380	2.1355e−09	5.7969	18,600	5.2781e−08	9.1875
	rsh-BGMRES-D-v1	1635	3.9361e−09	1.1094	2055	2.2047e−09	1.7656
	rsh-BGMRES-v2	9600	2.1677e−09	4.0938	11,300	2.3522e−09	5.9531
	rsh-BGMRES-D-v2	2100	3.6981e−09	1.5781	4810	3.0451e−09	5.0938
sherman4	rsh-BFOM	660	5.9177e−10	0.4531	1500	3.3237e−09	1.0156
	rsh-BFOM-D	330	2.7676e−10	0.4218	525	5.9144e−10	0.6250
	rsh-BGMRES-v1	660	2.6078e−09	0.5937	3000	2.9209e−06	2.2656
	rsh-BGMRES-D-v1	330	2.1252e−10	0.3906	525	3.3766e−10	0.6218
	rsh-BGMRES-v2	660	2.6406e−09	0.5156	1900	3.2142e−09	1.4219
	rsh-BGMRES-D-v2	735	6.9338e−09	0.8750	1110	6.5625e−08	1.6250
wathen120	rsh-BFOM	1680	1.9009e−08	28.297	1900	2.3336e−08	53.766
	rsh-BFOM-D	870	1.1184e−08	22.281	1205	1.3785e−08	44.891
	rsh-BGMRES-v1	1620	1.4964e−08	29.891	1900	9.6145e−09	58.594
	rsh-BGMRES-D-v1	825	1.2061e−08	19.000	1205	2.2034e−08	38.578
	rsh-BGMRES-v2	1620	1.3875e−08	29.797	1800	1.3131e−08	54.109
	rsh-BGMRES-D-v2	1260	2.6917e−08	25.922	1665	3.7816e−08	55.016
Dubcova2	rsh-BFOM	1020	8.7651e−09	51.641	600	2.0704e−08	33.922
	rsh-BFOM-D	645	1.9497e−08	45.000	610	1.3701e−09	42.000
	rsh-BGMRES-v1	960	2.2585e−08	50.016	600	2.3452e−08	35.188
	rsh-BGMRES-D-v1	645	5.7092e−09	41.078	525	2.8957e−08	33.016
	rsh-BGMRES-v2	960	9.4332e−07	49.531	600	1.9072e−08	34.797
	rsh-BGMRES-D-v2	840	1.3384e−07	47.891	740	3.5496e−07	46.172
bidiag	rsh-BFOM	-	-	-	-	-	-
	rsh-BFOM-D	510	2.4030e−10	1.7344	695	1.9708e−09	2.0469
	rsh-BGMRES-v1	-	-	-	-	-	-
	rsh-BGMRES-D-v1	510	2.6322e−10	1.3594	695	6.3707e−09	1.8281
	rsh-BGMRES-v2	6840	4.5704e−07	15.406	4900	2.5540e−06	11.125
	rsh-BGMRES-D-v2	3045	8.6032e−09	7.9531	3145	6.4237e−09	8.8750
tridiag	rsh-BFOM	-	-	-	-	-	-
	rsh-BFOM-D	510	6.9935e−10	1.7813	695	8.7645e−10	2.1094
	rsh-BGMRES-v1	-	-	-	-	-	-
	rsh-BGMRES-D-v1	465	2.2872e−09	1.1875	695	2.4902e−09	1.8438
	rsh-BGMRES-v2	-	-	-	6400	2.8854e−04	14.156
	rsh-BGMRES-D-v2	2205	9.7581e−09	5.7188	2775	1.6692e−08	7.6719



(a) $A =$ **nos5**. (b) $A =$ **sherman4**.

(c) $A =$ **wathen120**. (d) $A =$ **Dubcova2**.

(e) $A =$ **bidiag**. (f) $A =$ **tridiag**.

Fig. 1 The $\log_{10}$ plot of the maximum residual norms with respect to the number of matrix-vector products for example 2 with $s = 3$, $m = 20$, $k = 3$, and $l = 15$

coming from {9, 12}, and the value of the parameters k, s, and m is set to be 3, 3, and 20, respectively.

The numerical results for the different values of l are shown in Table 4. This table illustrates that the required matrix-vector products and CPU time for the deflated methods rsh-BFOM-D, rsh-BGMRES-D-v1, rsh-BGMRES-D-v2, and GMRES-DR-Sh are less than those of the non-deflated ones. Additionally, both rsh-BFOM-D and rsh-BGMRES-D-v1 methods achieve higher levels of performance compared with the rsh-BGMRES-D-v2 and GMRES-DR-Sh methods. It should be also noted that both rsh-BGMRES-D-v2 and GMRES-DR-Sh have almost the same performance with respect to the number of matrix-vector products and CPU time.

Table 4 Numerical results for example 3 with $s = 3$, $m = 20$, $k = 3$, and $l = 9, 12$

		nos5		memplus		sherman4		wathen120		Dubcova2	
	Method	mv	CPU	mv	CPU	mv	CPU	mv	CPU	mv	CPU
	rsh-BFOM	15,300	4.1406	360	2.1563	2940	1.4375	2520	39.328	540	26.016
	rsh-BGMRES-v1	13,620	4.7813	360	2.3125	2160	1.2656	2400	39.750	540	26.844
	rsh-BGMRES-v2	8580	3.1250	360	2.3594	2700	1.5313	2400	39.703	480	23.203
	GMRES-Sh	8520	2.9688	360	2.2813	2700	1.4375	1920	32.328	480	23.578
	rsh-BFOM-D	2508	1.5156	111	0.9062	417	0.3750	1182	24.984	315	18.156
$l = 9$	rsh-BGMRES-D-v1	2406	1.0625	111	0.8750	417	0.3438	1080	20.844	315	17.469
	rsh-BGMRES-D-v2	3663	1.8750	111	0.9375	1221	1.1719	1554	29.00	444	23.281
	GMRES-DR-Sh	3552	1.8750	111	0.9375	1221	1.1094	1443	26.172	444	23.250
	rsh-BFOM-D	2076	1.1875	108	0.8593	396	0.4375	1068	24.031	156	8.8906
$l = 12$	rsh-BGMRES-D-v1	1932	0.9531	108	0.8437	396	0.2969	1020	20.516	156	8.7656
	rsh-BGMRES-D-v2	2700	1.4844	108	0.9375	972	0.8438	1080	21.438	216	11.672
	GMRES-DR-Sh	3024	1.6875	108	0.8750	972	0.8438	1080	20.531	216	11.922

On the other hand, Table 4 shows that the number of matrix-vector products or CPU time for the deflated methods decreases when the value of l increases from 9 to 12. In specific, for $l = 12$, the rsh-BGMRES-D-v1 method outperforms the other methods with respect to the number of matrix-vector products and CPU time.

Example 4 In this example, we take the $s \times s$ diagonal matrix $\Sigma_j = \text{diag}([\sigma_j^{(1)}, \ldots, \sigma_j^{(s)}])$ as $\Sigma_j = \alpha_j I_s$, where $j = 1, 2, \ldots, k$. In fact, we consider the special case $\sigma_j^{(i)} = \alpha_j$, for $i = 1, \ldots, s$. Moreover, we choose $\alpha_j = -8*(j+1)*10^{-8}$. The test matrices used in this example are **raefsky1**, **poisson3Da**, **wathen100**, and **poisson3Db**. We also take $m = 20$, $k = 3$, $s = 3$, and $l = 3$.

The numerical results for the mentioned test matrices are reported in Table 5. Except for the test matrix **wathen100**, the numerical results of this table demonstrate that the rsh-BFOM method is superior to the rsh-BGMRES-v1 and rsh-BGMRES-v2 methods. We can also see that the number of matrix-vector products and CPU time for the deflated version of the rsh-BFOM, rsh-BGMRES-v1, and rsh-BGMRES-v2 methods are smaller than those of the non-deflated ones. In particular, for the test matrix **raefsky1**, the rsh-BFOM, rsh-BGMRES-v1, and rsh-BGMRES-v2 methods do not converge, whereas the performance of their deflated versions is quite satisfactory. Table 5 also indicates that rsh-BGMRES-D-v1 outperforms the other methods in terms of both the number of matrix-vector products and CPU time.

For the sake of studying the numerical behavior of the different methods, Fig. 2 shows the $\log_{10}$ plot of the maximum residual norms with respect to the number of matrix-vector products. In some cases, e.g., **poisson3Da** and **poisson3Db**, it should be mentioned that the curves of rsh-BGMRES-v1 and rsh-BGMRES-v2

Table 5 Numerical results for example 4 with $s = 3$, $m = 20$, $k = 3$, and $l = 3$

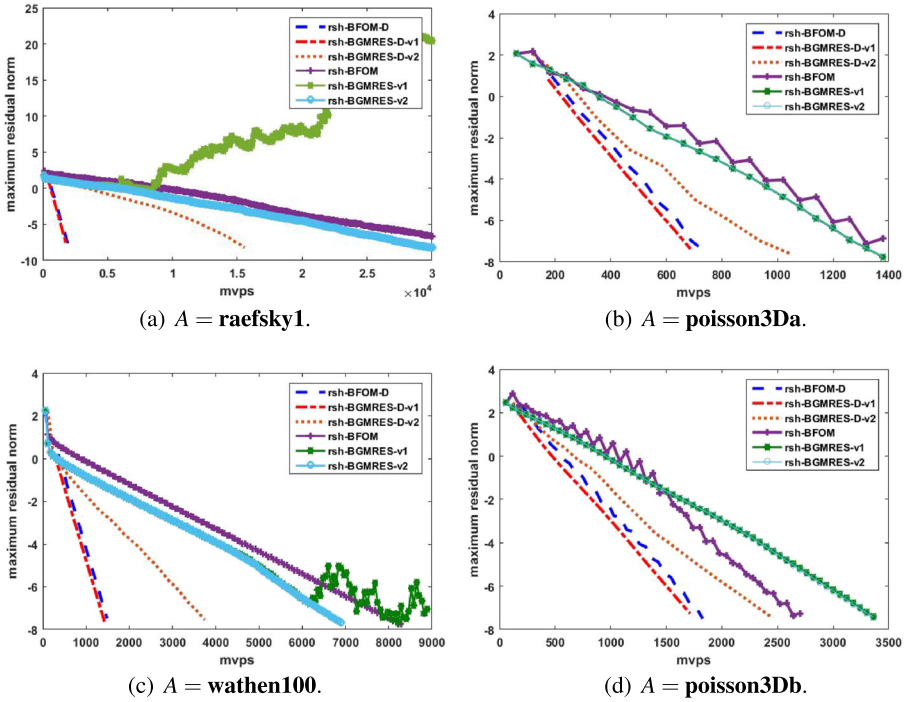
Method	mv	mnres	CPU	mv	mnres	CPU
	raefsky1			poisson3Da		
rsh-BFOM	-	-	-	1380	8.8852e−09	8.3906
rsh-BFOM-D	1998	2.5506e−09	8.3906	744	1.4904e−09	6.2344
rsh-BGMRES-v1	-	-	-	1380	6.9788e−09	9.6719
rsh-BGMRES-D-v1	1827	6.8562e−09	6.0156	687	5.7101e−09	5.1406
rsh-BGMRES-v2	-	-	-	1380	6.9724e−09	9.7344
rsh-BGMRES-D-v2	15,561	6.3848e−09	49.313	1053	2.2605e−08	8.0625
	wathen100			poisson3Db		
rsh-BFOM	8280	1.4089e−08	109.08	2700	1.2945e−08	186.94
rsh-BFOM-D	1485	1.4603e−08	26.000	1827	2.0786e−08	158.75
rsh-BGMRES-v1	8880	4.2202e−07	131.91	3360	2.6493e−08	247.64
rsh-BGMRES-D-v1	1428	9.2839e−09	22.922	1713	2.4723e−08	139.89
rsh-BGMRES-v2	6900	2.3093e−08	103.91	3360	2.4485e−08	248.55
rsh-BGMRES-D-v2	3744	2.8557e−08	58.313	2457	3.3705e−08	186.97

are overlapped with each other. According to this figure, the rsh-BFOM-D, rsh-BGMRES-D-v1, and rsh-BGMRES-D-v2 methods converge much faster than their non-deflated versions.

Example 5 In the current example, a comparison is drawn between the performance results of our proposed methods and those of GMRES-Sh [33], BGMRES-Sh [19], GMRES-DR-Sh [33], and BGMRES-DR-Sh [45]. We use the four test matrices **sherman4**, **raefsky1**, **wathen100**, and **wathen120**. Here, the values of l, k, s, and m are set as 9, 3, 3, and 20, respectively. Moreover, similar to example 4, the diagonal matrices Σ_j are taken as $\Sigma_j = \alpha_j I_s$ for $j = 1, 2, \ldots, k$. In this regard, we consider two cases for α_j as follows:

- Case 1: The values of α_j are selected randomly using a uniform distribution on $[-3, -1]$,
- Case 2: The values of α_j are taken from $\{-1, -10, -100\}$.

The numerical results for the two abovementioned choices of shifts are collected in Table 6. It can be observed from this table that the use of the deflation technique into the method "rsh-BGMRES with an unshifted base system (i.e., rsh-BGMRES-v1)" is particularly effective to enhance the efficiency level of it, while in most cases, the performance results of the deflated version of rsh-BFOM, rsh-BGMRES-v2, GMRES-Sh, and BGMRES-Sh are inferior to those of their corresponding non-deflated ones. It can be also inferred that in almost all cases, the proposed rsh-BGMRES-D-v1 method achieves the best level of performance among other methods in terms of the number of matrix-vector products or CPU time. Specifically, the rsh-BGMRES-D-v1 works better than the other deflated methods. For instance, we



(a) $A =$ **raefsky1**. (b) $A =$ **poisson3Da**.

(c) $A =$ **wathen100**. (d) $A =$ **poisson3Db**.

Fig. 2 The $\log_{10}$ plot of the maximum residual norms with respect to the number of matrix-vector products for example 4 with $s = 3$, $m = 20$, $k = 3$, and $l = 3$

can see that for the test matrices **wathen100** and **wathen120**, the rsh-BGMRES-D-v2, GMRES-DR-Sh, and BGMRES-DR-Sh methods are not convergent within 501 restarts, whereas the rsh-BGMRES-D-v1 method presents a very favorable outcome. It is also worth noting that the numerical results indicate that except for the test matrices **wathen100** and **wathen120**, the rsh-BGMRES-D-v2 method produces superior results with respect to the CPU time in comparison with the GMRES-DR-Sh and BGMRES-DR-Sh methods.

Example 6 The aim of this example is to consider the quantum chromodynamics (QCD) application to illustrate the performance of the rsh-BFOM, rsh-BGMRES-v1, and rsh-BGMRES-v2 methods and their deflated versions. It is known that the QCD application consists of problems with multiple right-hand sides and multiple shifts [33]. In this example, a set of six QCD matrices, denoted by D_i for $i = 1, \ldots, 6$, is chosen from the University of Florida Sparse Matrix Library. For each matrix D_i, the base matrix is considered as $A_i = (\frac{1}{\kappa_c} + 10^{-3})I - D_i$ in which the parameter κ_c represents the critical value. Furthermore, the right-hand side is set as $B = \texttt{ones}(n, s)$. Table 7 describes some details of these QCD matrices. It should be also mentioned that the values of κ_c are considered as suggested in [65].

Table 8 reports the number of matrix-vector products and CPU time when $m = 120$, $k = 2$, $s = 3$, and $l = 3$. In each case, except for A_2 and A_4, this table shows

Table 6 Numerical results for example 5 with $s = 3$, $m = 20$, $k = 3$, and $l = 9$

		sherman4		raefsky1		wathen100		wathen120	
α_j	Method	mv	CPU	mv	CPU	mv	CPU	mv	CPU
	rsh-BFOM	300	0.2188	120	0.3281	780	9.9219	780	12.625
	rsh-BGMRES-v1	300	0.2031	120	0.3594	780	9.8750	780	13.000
	rsh-BGMRES-v2	300	0.2188	120	0.3594	780	10.000	780	13.234
	GMRES-Sh	300	0.2344	120	0.3125	780	9.7969	780	13.234
Case 1	BGMRES-Sh	300	0.2500	120	0.3594	780	9.8125	780	13.047
	rsh-BFOM-D	264	0.2031	110	0.3594	621	10.281	672	14.125
	rsh-BGMRES-D-v1	213	0.2031	110	0.2813	621	8.7031	621	11.844
	rsh-BGMRES-D-v2	333	0.2969	110	0.3594	-	-	-	-
	GMRES-DR-Sh	333	0.2969	110	0.3750	-	-	-	-
	BGMRES-DR-Sh	417	0.7344	110	0.7188	-	-	-	-
	rsh-BFOM	300	0.2031	120	0.3125	600	7.0625	540	8.3750
	rsh-BGMRES-v1	300	0.2031	120	0.3438	600	8.2031	540	9.2500
	rsh-BGMRES-v2	300	0.2031	120	0.3125	600	7.8594	540	9.4219
	GMRES-Sh	300	0.2500	120	0.3594	600	8.3438	540	9.2188
Case 2	BGMRES-Sh	300	0.2656	120	0.3906	600	8.1094	540	9.3750
	rsh-BFOM-D	213	0.2031	111	0.3594	519	8.4844	468	9.7344
	rsh-BGMRES-D-v1	213	0.1718	111	0.2813	519	7.6875	468	9.0781
	rsh-BGMRES-D-v2	333	0.2813	111	0.3281	-	-	-	-
	GMRES-DR-Sh	333	0.3438	111	0.3438	-	-	-	-
	BGMRES-DR-Sh	366	0.7188	111	0.7031	-	-	-	-

that the rsh-BFOM method uses the least CPU time compared with other methods. In addition, rsh-BGMRES-v2 produces better results than rsh-BGMRES-v1 in all the cases. The results obtained in this table indicate that rsh-BFOM-D requires fewer matrix-vector products or CPU time than rsh-BGMRES-D-v1 and rsh-BGMRES-D-v2. We can also see that, except for A_1, the number of matrix-vector products for the rsh-BFOM-D, rsh-BGMRES-D-v1, and rsh-BGMRES-D-v2 methods is either

Table 7 Information of the six QCD matrices used in example 6

Matrix ID	QCD matrix	Size	Nonzeros	Type	κ_c
A_1	conf5_0-4x4-14	3072	119,808	Complex	0.20328
A_2	conf5_0-4x4-22	3072	119,808	Complex	0.20235
A_3	conf5_0-4x4-26	3072	119,808	Complex	0.21070
A_4	conf5_4-8x8-10	49,152	1,916,928	Complex	0.17843
A_5	conf6_0-8x8-20	49,152	1,916,928	Complex	0.15717
A_6	conf6_0-8x8-80	49,152	1,916,928	Complex	0.15623

Table 8 Numerical results for example 6 with $s = 3$, $m = 120$, $k = 2$, and $l = 3$

Method	mv	CPU	mv	CPU	mv	CPU
	A_1		A_2		A_3	
rsh-BFOM	360	8.4219	720	16.766	360	8.6094
rsh-BGMRES-v1	360	10.797	720	21.672	720	19.469
rsh-BGMRES-v2	360	8.8906	720	15.563	720	15.359
rsh-BFOM-D	360	8.5000	360	8.1719	360	8.2344
rsh-BGMRES-D-v1	708	24.125	708	24.453	708	23.859
rsh-BGMRES-D-v2	708	21.688	708	21.344	780	21.938
	A_4		A_5		A_6	
rsh-BFOM	720	321.22	720	320.20	360	159.47
rsh-BGMRES-v1	720	326.00	720	326.25	720	323.75
rsh-BGMRES-v2	720	318.38	720	323.94	360	161.06
rsh-BFOM-D	708	325.56	708	326.64	360	159.65
rsh-BGMRES-D-v1	708	331.63	708	332.14	708	330.66
rsh-BGMRES-D-v2	708	328.64	708	325.45	708	330.80

smaller than or equal to that of their corresponding non-deflated ones, while the CPU time required by the deflated methods is slightly higher than that required by their corresponding non-deflated ones. Therefore, this experiment shows that the rsh-BFOM, rsh-BGMRES-v1, and rsh-BGMRES-v2 methods perform better than their deflated versions in terms of CPU time.

6 Conclusion

Frommer and Gl*ä*ssner [25] and Simoncini [27] have proposed the restarted shifted GMRES and restarted shifted FOM methods, respectively. In this paper, due to the shift-invariance property of the block subspace and the collinearity property, we have established two block versions for restarted shifted FOM and restarted shifted GMRES, called rsh-BFOM and rsh-BGMRES, so as to solve sequences of shifted linear systems. We have presented two implementations of the rsh-BGMRES method, i.e., "rsh-BGMRES with an unshifted base system" and "rsh-BGMRES with a variable shifted base system." Moreover, we have introduced the deflated version of the abovementioned methods. The numerical experiments on several large and sparse test matrices have demonstrated that (1) the rsh-BFOM and rsh-BGMRES methods are efficient for solving sequences of shifted linear systems; (2) rsh-BFOM outperforms "rsh-BGMRES with a variable shifted base system" for some cases, while "rsh-BGMRES with a variable shifted base system" works better than rsh-BFOM for some other cases; (3) the performance of rsh-BFOM and "rsh-BGMRES with a variable shifted base system" is better than that of "rsh-BGMRES with an unshifted base system"; (4) the application of the deflation technique into the rsh-BFOM and

rsh-BGMRES methods can boost the efficiency level of the original methods; (5) the incorporation of the deflation technique into both "the restarted shifted block FOM method" and "the rsh-BGMRES with a fixed base system" produces better results in comparison with the combination of the deflation technique and "the rsh-BGMRES with a variable base system"; (6) the deflation version of "rsh-BGMRES with an unshifted base system" outperforms the other comparison methods. However, from the experiment on the QCD application, we have found that the number of matrix-vector products for the methods rsh-BFOM, "rsh-BGMRES with an unshifted base system" and "rsh-BGMRES with a variable shifted base system" can be smaller than or equal to that of their corresponding deflated methods, while the CPU time demanded by the deflated methods was slightly higher compared with the time required for their corresponding non-deflated methods. Therefore, further investigations are needed to enhance the performance of the deflated version of the proposed methods. We end this conclusion by pointing out that a recent work [43] based on the use of *-algebras described in [42] could help describe a general formalism for the resolution of shifted linear systems. It will also certainly contribute to better understand and analyze the convergence of these methods. We hope that this could be achieved in a future work. We also intend to apply the idea of preconditioning strategy to deflated variant of the rsh-BFOM and rsh-BGMRES methods.

References

1. Datta, B.N.: Numerical Methods for Linear Control Systems. Academic Press, 82 (2004)
2. Laub, A.: Numerical linear algebra aspects of control design computations. IEEE Trans. Autom. Control **30**(2), 97–108 (1985). https://doi.org/10.1109/TAC.1985.1103900
3. Bloch, J., Frommer, A., Lang, B., Wettig, T.: An iterative method to compute the sign function of a non-Hermitian matrix and its application to the overlap Dirac operator at nonzero chemical potential. Comput. Phys. Commun. **177**(12), 933–943 (2007). https://doi.org/10.1016/j.cpc.2007.07.012, http://www.sciencedirect.com/science/article/pii/S0010465507003633
4. Darnell, D., Morgan, R.B., Wilcox, W.: Deflation of eigenvalues for iterative methods in lattice QCD. Nuclear Physics B - Proceedings Supplements **129-130**, 856–858 (2004). https://doi.org/10.1016/S0920-5632(03)02734-8, http://www.sciencedirect.com/science/article/pii/S0920563203027348
5. Freund, R.W.: Solution of shifted linear systems by quasi-minimal residual iterations. Numerical Linear Algebra. Berlin, Boston: De Gruyter, pp 101–121 https://doi.org/10.1515/9783110857658.101 (1993)
6. Narayanan, R., Neuberger, H.: An Alternative to domain wall fermions. Phys. Rev. **D62**, 074504 (2000). https://doi.org/10.1103/PhysRevD.62.074504. hep-lat/0005004
7. Sakurai, T., Tadano, H., Kuramashi, Y.: Application of block Krylov subspace algorithms to the Wilson-Dirac equation with multiple right-hand sides in lattice QCD. Comput. Phys. Commun. **181**(1), 113–117 (2010). https://doi.org/10.1016/j.cpc.2009.09.006, http://www.sciencedirect.com/science/article/pii/S0010465509002859
8. Gallopoulos, E., Saad, Y.: Efficient solution of parabolic equations by Krylov approximation methods. SIAM J. Sci. Stat. Comput. **13**(5), 1236–1264 (1992). https://doi.org/10.1137/0913071
9. Garrappa, R., Popolizio, M.: On the use of matrix functions for fractional partial differential equations. Math. Comput. Simul. **81**(5), 1045–1056 (2011). https://doi.org/10.1016/j.matcom.2010.10.009, http://www.sciencedirect.com/science/article/pii/S0378475410003150, Important aspects on structural dynamical systems and their numerical computation
10. Meerbergen, K.: The solution of parametrized symmetric linear systems. SIAM Journal on Matrix Analysis and Applications **24**(4), 1038–1059 (2003). https://doi.org/10.1137/S0895479800380386

11. Weideman, J.A.C., Trefethen, L.N.: Parabolic and Hyperbolic Contours for computing the Bromwich integral. Math. Comput. **76**(259), 1341–1356 (2007). http://www.jstor.org/stable/40234432
12. Ikegami, T., Sakurai, T.: Contour integral eigensolver for non-Hermitian systems: A Rayleigh-Ritz-type approach. Taiwanese J. Math. **14**(3A), 825–837 (2010). https://doi.org/10.11650/twjm/1500405869. https://doi.org/10.11650/twjm/1500405869
13. Ikegami, T., Sakurai, T., Nagashima, U.: A filter diagonalization for generalized eigenvalue problems based on the SakuraiSugiura projection method. J. Comput. Appl. Math. **233**(8), 1927–1936 (2010)
14. Feriani, A., Perotti, F., Simoncini, V.: Iterative system solvers for the frequency analysis of linear mechanical systems. Comput. Methods Appl. Mech. Eng. **190**(13), 1719–1739 (2000). https://doi.org/https://doi.org/10.1016/S0045-7825(00)00187-0. http://www.sciencedirect.com/science/article/pii/S0045782500001870
15. Saibaba, A., Bakhos, T., Kitanidis, P.: A flexible Krylov solver for shifted systems with application to Oscillatory Hydraulic Tomography. SIAM J. Sci. Comput. **35**(6), A3001–A3023 (2013). https://doi.org/10.1137/120902690
16. Sogabe, T., Hoshi, T., L., Z.S., Fujiwara, T.: On a weighted quasi-residual minimization strategy for solving complex symmetric shifted linear systemss. Electron. Trans. Numer. Anal. **31**, 126–140 (2008)
17. Sogabe, T., Hoshi, T., Zhang, S.-L., Fujiwara, T.: A numerical method for calculating the Green's function arising from structure theory. In: Kaneda, Y., Kawamura, H., Sasai, M. (eds.) Frontiers of Computational Science, Springer Berlin Heidelberg, Berlin, Heidelberg, pp 189–195 (2007)
18. Takayama, R., Hoshi, T., Sogabe, T., Zhang, S.-L., Fujiwara, T.: Linear algebraic calculation of the Green's function for large-scale electronic structure theory. Phys. Rev. B **73**, 165108 (2006). https://doi.org/10.1103/PhysRevB.73.165108. https://link.aps.org/doi/10.1103/PhysRevB.73.165108
19. Wu, G., Wang, Y., Jin, X.: A preconditioned and shifted GMRES algorithm for the PageRank problem with multiple damping factors. SIAM J. Sci. Comput. **34**(5), A2558–A2575 (2012). https://doi.org/10.1137/110834585
20. Elbouyahyaoui, L., Heyouni, M., Jbilou, K., Messaoudi, A.: A block Arnoldi method for the solution of the Sylvester-observer equations. Electron. Trans. Numer. Anal. **47**, 18–36 (2017)
21. Benzi, M., Bertaccini, D.: Approximate inverse preconditioning for shifted linear systems. BIT Numer. Math. **43**(2), 231–244 (2003). https://doi.org/10.1023/A:1026089811044
22. Dehghan, M., Mohammadi-Arani, R.: Generalized product-type methods based on bi-conjugate gradient (GPBiCG) for solving shifted linear systems. Comput. Appl. Math. **36**(4), 1591–1606 (2017). https://doi.org/10.1007/s40314-016-0315-y
23. Frommer, A.: BiCGStab(*l*) for families of shifted linear systems. Computing **70**(2), 87–109 (2003). https://doi.org/10.1007/s00607-003-1472-6
24. van den Eshof, J., Sleijpen, G.L.G.: Accurate conjugate gradient methods for families of shifted systems. Appl. Numer. Math. **49**(1), 17–37 (2004). https://doi.org/10.1016/j.apnum.2003.11.010, http://www.sciencedirect.com/science/article/pii/S0168927403001934, Numerical Algorithms, Parallelism and Applications
25. Frommer, A., Glässner, U.: Restarted GMRES for shifted linear systems. SIAM J. Sci. Comput. **19**(1), 15–26 (1998). https://doi.org/10.1137/S1064827596304563
26. Jing, Y.-F., Yuan, P., Huang, T.-Z.: A simpler GMRES and its adaptive variant for shifted linear systems. Numerical Linear Algebra with Applications **24**(1), e2076 (2017). https://doi.org/10.1002/nla.2076. e2076 nla.2076
27. Simoncini, V.: Restarted full orthogonalization method for shifted linear systems. BIT Numer. Math. **43**(2), 459–466 (2003). https://doi.org/10.1023/A:1026000105893
28. Soodhalter, K.M., Szyld, D.B., Xue, F.: Krylov subspace recycling for sequences of shifted linear systems. Appl. Numer. Math. **81**, 105–118 (2014). https://doi.org/10.1016/j.apnum.2014.02.006, http://www.sciencedirect.com/science/article/pii/S0168927414000208
29. Gu, X.-M., Huang, T.-Z., Carpentieri, B., Imakura, A., Zhang, K., Du, L.: Efficient variants of the CMRH method for solving a sequence of multi-shifted non-Hermitian linear systems simultaneously. arXiv e-prints, arXiv:1611.00288 (2016)
30. Gu, X.-M., Huang, T.-Z., Yin, G., Carpentieri, B., Wen, C., Du, L.: Restarted Hessenberg method for solving shifted nonsymmetric linear systems. J. Comput. Appl. Math. **331**, 166–177 (2018)
31. Ahmad, M., Szyld, D., van Gijzen, M.: Preconditioned multishift BiCG for $\mathcal{H}_2$-optimal model reduction. SIAM Journal on Matrix Analysis and Applications **38**(2), 401–424 (2017). https://doi.org/10.1137/130914905

32. Soodhalter, K.M.: Two recursive GMRES-type methods for shifted linear systems with general preconditioning. Electron. Trans. Numer. Anal. **451**, 499–523 (2016)
33. Darnell, D., Morgan, R.B., Wilcox, W.: Deflated GMRES for systems with multiple shifts and multiple right-hand sides. Linear Algebra Appl. **429**(10), 2415–2434 (2008). https://doi.org/10.1016/j.laa.2008.04.019, http://www.sciencedirect.com/science/article/pii/S0024379508001997, Special Issue in honor of Richard S. Varga
34. Yin, J.-F., Yin, G.-J.: Restarted full orthogonalization method with deflation for shifted linear systems. Numerical Mathematics: Theory, Methods and Applications **7**(3), 399–412 (2014). https://doi.org/10.1017/S1004897900000179
35. Gu, G.: Restarted GMRES augmented with harmonic Ritz vectors for shifted linear systems. Int. J. Comput. Math. **82**(7), 837–849 (2005). https://doi.org/10.1080/00207160512331323317
36. Simoncini, V.: On the numerical solution of $A X - X B = C$. BIT Numer. Math. **36**(4), 814–830 (1996). https://doi.org/10.1007/BF01733793
37. El Guennouni, A., Jbilou, K., Riquet, A.J.: Block Krylov subspace methods for solving large Sylvester equations. Numerical Algorithms **29**(1), 75–96 (2002). https://doi.org/10.1023/A:1014807923223
38. Robbé, M., Sadkane, M.: A convergence analysis of GMRES and FOM methods for Sylvester equations. Numerical Algorithms **30**(1), 71–89 (2002). https://doi.org/10.1023/A:1015615310584
39. Jbilou, K.: Low rank approximate solutions to large Sylvester matrix equations. Appl. Math. Comput. **177**(1), 365–376 (2006)
40. Robbé, M., Sadkane, M.: Use of near-breakdowns in the block Arnoldi method for solving large Sylvester equations. Appl. Numer. Math. **58**(4), 486–498 (2008). https://doi.org/https://doi.org/10.1016/j.apnum.2007.01.025, http://www.sciencedirect.com/science/article/pii/S0168927407000372, Selected Papers from the Seventh IMACS International Symposium on Iterative Methods in Scientific Computing
41. Datta, B.N., Saad, Y.: Arnoldi methods for large Sylvester-like observer matrix equations, and an associated algorithm for partial spectrum assignment. Linear Algebra Appl. **154-156**, 225–244 (1991). https://doi.org/10.1016/0024-3795(91)90378-A, http://www.sciencedirect.com/science/article/pii/002437959190378A
42. Frommer, A., Lund, K., Szyld, D.B.: Block Krylov subspace methods for functions of matrices. Electron. Trans. Numer. Anal. **47**, 100–126 (2017)
43. Kubínová, M., Soodhalter, K.M.: Admissible and attainable convergence behavior of block Arnoldi and GMRES. https://arxiv.org/abs/1907.03677 (2019)
44. Soodhalter, K.: Block Krylov subspace recycling for shifted systems with unrelated right-hand sides. SIAM J. Sci. Comput. **38**(1), A302–A324 (2016). https://doi.org/10.1137/140998214
45. Sun, D.-L., Huang, T.-Z., Jing, Y.-F., Carpentieri, B.: A block GMRES method with deflated restarting for solving linear systems with multiple shifts and multiple right-hand sides. Numerical Linear Algebra with Applications **25**(5), e2148 (2018)
46. Wu, G., Pang, H.-K., Sun, J.-L.: A shifted block FOM algorithm with deflated restarting for matrix exponential computations. Appl. Numer. Math. **127**, 306–323 (2018). https://doi.org/10.1016/j.apnum.2018.01.015, http://www.sciencedirect.com/science/article/pii/S0168927418300242
47. Simoncini, V., Gallopoulos, E.: Convergence properties of block GMRES and matrix polynomials. Linear Algebra Appl. **247**, 97–119 (1996). https://doi.org/10.1016/0024-3795(95)00093-3, http://www.sciencedirect.com/science/article/pii/0024379595000933
48. Gaul, A.: Recycling krylov subspace methods for sequences of linear systems analysis and applications. Ph.D. Thesis (2014)
49. Parks, M.L., de Sturler, E., Mackey, G., Johnson, D.D., Maiti, S.: Recycling krylov subspaces for sequences of linear systems. SIAM J. Sci. Comput. **28**(5), 1651–1674 (2006). https://doi.org/10.1137/040607277
50. Morgan, R.: GMRES with deflated restarting. SIAM J. Sci. Comput. **24**(1), 20–37 (2002). https://doi.org/10.1137/S1064827599364659
51. Morgan, R.B.: Restarted block-GMRES with deflation of eigenvalues. Appl. Numer. Math. **54**(2), 222–236 (2005)
52. Gaul, A., Gutknecht, M.H., Liesen, J., Nabben, R.: A framework for deflated and augmented Krylov subspace methods. SIAM Journal on Matrix Analysis and Applications **34**(2), 495–518 (2013)
53. Giraud, L., Gratton, S., Pinel, X., Vasseur, X.: Flexible GMRES with deflated restarting. SIAM J. Sci. Comput. **32**(4), 1858–1878 (2010)

54. Meng, J., Zhu, P.-Y., Li, H.-B., Gu, X.-M.: A deflated block flexible GMRES-DR method for linear systems with multiple right-hand sides. Electron. Trans. Numer. Anal. **41**, 478–496 (2014)
55. de Sturler, E.: Nested Krylov methods based on GCR. J. Comput. Appl. Math. **67**(1), 15–41 (1996)
56. Gutknecht, M.H.: Block krylov space methods for linear systems with multiple right-hand sides: An introduction (2007)
57. Saad, Y.: Iterative methods for sparse linear systems. Society for Industrial and Applied Mathematics, 2nd edition. https://epubs.siam.org/doi/abs/10.1137/1.9780898718003 (2003)
58. Kent, M.D.: Chebyshev, Krylov, Lanczos: Matrix relationships and computations. Ph.D. Thesis, Stanford University, Stanford, CA, USA. UMI Order No: GAX89-25896 (1989)
59. Vital, B.: Etude de quelques méthodes de résolution de problémes linéaires de grande taille sur multiprocesseur. Ph.D. Thesis, Université Rennes 1. http://www.sudoc.fr/044024320 (1990)
60. Golub, G., Nash, S., Van Loan, C.: A Hessenberg-Schur method for the problem $A X + X B = C$. IEEE Trans. Autom. Control **24**(6), 909–913 (1979). https://doi.org/10.1109/TAC.1979.1102170
61. Chan, T., Wan, W.: Analysis of projection methods for solving linear systems with multiple right-hand sides. SIAM J. Sci. Comput. **18**(6), 1698–1721 (1997). https://doi.org/10.1137/S1064827594273067
62. Simoncini, V., Gallopoulos, E.: An iterative method for nonsymmetric systems with multiple right-hand sides. SIAM J. Sci. Comput. **16**(4), 917–933 (1995). https://doi.org/10.1137/0916053
63. Davis, T.A., Hu, Y.: The University of Florida sparse matrix collection. ACM Transactions on Mathematical Software (TOMS) **38**(1), 1 (2011)
64. Morgan, R.B.: A restarted GMRES method augmented with eigenvectors. SIAM Journal on Matrix Analysis and Applications **16**(4), 1154–1171 (1995)
65. Zhong, H.-X., Gu, X.-M.: A flexible and adaptive Simpler GMRES with deflated restarting for shifted linear systems. Computers & Mathematics with Applications **78**(3), 997–1007 (2019)

Affiliations

Lakhdar Elbouyahyaoui[1] · Mohammed Heyouni[2] · Azita Tajaddini[3] · Farid Saberi-Movahed[4]

Lakhdar Elbouyahyaoui
lakhdarr2000@yahoo.fr

Azita Tajaddini
azita_tajaddinii@yahoo.com

Farid Saberi-Movahed
f.saberimovahed@kgut.ac.ir; fdsaberi@gmail.com

1 Centre Régional des Métiers de l'Education et de la Formation, Fes, Morocco

2 L.M.P.A, Université du Littoral Côte d'Opale, 50 rue F. Buisson BP. 699, F-62228 Calais Cedex, Dunkirk, France

3 Department of Applied Mathematics, Faculty of Mathematics and Computer, Shahid Bahonar University of Kerman, Kerman, Iran

4 Department of Applied Mathematics, Faculty of Sciences and Modern Technologies, Graduate University of Advanced Technology, Kerman, Iran