Journal of Scientific Computing (2021) 86:9
https://doi.org/10.1007/s10915-020-01376-w

Two New Variants of the Simpler Block GMRES Method with Vector Deflation and Eigenvalue Deflation for Multiple Linear Systems

Azita Tajaddini[1] · Gang Wu[2] · Farid Saberi-Movahed[3] · Najmeh Azizizadeh[1]

Received: 13 August 2019 / Revised: 24 October 2020 / Accepted: 27 November 2020 /
Published online: 2 January 2021

Abstract
In this paper, two effective methods based on the simpler block GMRES method are established in order to solve the linear systems of equations with multiple right-hand sides. The first method is derived from the simpler block GMRES method with vector deflation restarting (SBGMRES-DR). The second method is constructed from a combination of SBGMRES-DR with the eigenvalue deflation technique, which is called the deflated simpler block GMRES method with vector deflation restarting (D-SBGMRES-DR). To be more specific, SBGMRES-DR is capable of removing linearly or almost linearly dependent vectors created by the block Arnoldi process. On the other hand, D-SBGMRES-DR not only deletes linearly or almost linearly dependent vectors but also retains harmonic Ritz vectors associated with the smallest harmonic Ritz values in magnitude, and adds them to the new search subspace at the time of restart. Finally, a wide range of practical experiments are carried out to assess the efficiency of the proposed methods. The numerical results indicate that the D-SBGMRES-DR method outperforms the compared methods with respect to the number of matrix–vector products and the computational time.

Keywords Simpler block GMRES · Multiple linear systems · Vector deflation · Eigenvalue deflation.

Mathematics Subject Classification 65F15 · 65F10

✉ Azita Tajaddini
atajadini@uk.ac.ir

[1] Department of Applied Mathematics, Faculty of Mathematics and Computer, Shahid Bahonar University of Kerman, Kerman, Iran

[2] School of Mathematics, China University of Mining and Technology, Xuzhou 221116, Jiangsu, People's Republic of China

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

1 Introduction

In this paper, we are concerned with the solution of the linear systems with multiple right-hand sides

$$AX = B, \tag{1}$$

where $A \in \mathbb{C}^{n\times n}$ is a large nonsingular matrix, $B \in \mathbb{C}^{n\times p}$ is of full rank, and $X \in \mathbb{C}^{n\times p}$ with $p \ll n$. Such problems are at the center of consideration in a large number of applications, for example electromagnetic [37], model reduction in circuit simulation [15], quantum chromodynamics (QCD) [3,4,34], dynamics of structures [11] and other applied problems.

For more than two decades, a great many block Krylov subspace methods have been proposed for the solution of (1). Based on a discussion presented by Frommer et al. [16], block Krylov subspace methods can be divided into three main categories: classical block methods [21,22], global block methods [6,23] and loop-interchange block methods [17,31]. With this regard, Frommer et al. [17] have taken into consideration three particular types of the block inner product, i.e., classical, global and loop-interchange, to provide a general formalism for some block Krylov subspace methods such as the block FOM, block GMRES and block Radau–Arnoldi methods. Among the most prominent instances of block Krylov subspace methods are the block GMRES (BGMRES) method and its variants, which have gained special attention by researchers; we refer to [17,20,28,33,35,36] and the references therein for more supplementary information on this topic.

As a successful variant of the BGMRES method, it can be pointed to the simpler block GMRES (SBGMRES) method [27], which is a block version of the simpler GMRES (SGMRES) method established by Walker and Zhou [41]. More recently, motivated by the idea of SGMRES, Abdaoui et al. [1] have described the simpler block CMRH (SBCMRH) method, which exploits a modified version of the block Hessenberg process at a lower cost to produce a basis for the block Krylov subspace and achieves the favorable performance results in comparison with SBGMRES. A remarkable advantage of the SBGMRES method is to avoid the factorization of a block upper Hessenberg matrix and to alleviate arithmetic operations in comparison with BGMRES [10,41]. However, the lack of numerical stability can be cited as one of the drawbacks of both the SGMRES and SBGMRES methods. In order to address this issue, Jiránek et al. [25] have presented an adaptive simpler GMRES algorithm (Ad-SGMRES). Then, Zhong et al. [46] have proposed a flexible and adaptive simpler block GMRES with deflated restarting (FAd-SBGMRES-DR), and additionally they have studied in-depth the superiority of FAd-SBGMRES-DR over SBGMRES from practical viewpoints.

In order to improve the convergence of the BGMRES method, Morgan [30] has conducted a research on the crucial role of small eigenvalues in the convergence of restarted BGMRES and has introduced a new version of the restarted BGMRES with the deflation of eigenvalues (BGMRES-DR) in which the principle aim is to eliminate small eigenvalues at each restart. With getting inspiration from the deflation procedure proposed in [30], Meng et al. [29] have established a flexible version of the BGMRES with deflation of eigenvalues to address the occurrence of the possible linear dependence of some columns of the block residuals at each iteration of the block Arnoldi process. As another example, Al Daas et al. [2] have presented a novel version of BGMRES, called the enlarged GMRES (EGMRES) method. The central idea of EGMRES is to expand the block Krylov subspace by using a number of new basis vectors obtained for each right hand-side. Indeed, EGMRES makes use of two strategies: the first is based on detecting inexact breakdowns, and the second is based on the deflation of eigenvalues. Sun et al. [38] have introduced an effective scheme for BGMRES, which integrates simultaneously two notions "eigenspace recycling" and "initial deflation" in order to boost the performance of BGMRES. In recent years, the conjugate gradient (CG)-like methods according to the deflation strategy have been also employed efficiently in symmetric positive definite systems with multiple right-hand sides. For example, Ji and Li [24] have first studied the subject of the rank deficiency that leads to the breakdown in the block CG method and then offered a breakdown-free block CG method in order to enhance the effectiveness of the block CG method. Xiang et al. [43] have designed a powerful formulation for the deflated block CG approach, which succeeds in the challenge of maintaining the orthogonality between the block residual vectors and the deflation subspace.

It is worthwhile to mention that an important kind of the deflation technique is the vector deflation. The main purpose of the vector deflation is to detect a possible linear dependence during the block iterative process and to remove linearly or almost linearly dependent vectors from the subspace [29]. Despite the fact that the vector deflation technique improves the convergence of block Krylov subspace methods and reduces their computational works [29,30,32], it may result in a slow convergence rate because of the loss of information [26]. In connection with the BGMRES and SBGMRES methods, remarkable efforts have been devoted to the efficient use of the vector deflation. For instance, Robbé and Sadkane [32] have introduced a modified version of the block Arnoldi with the vector deflation for BGMRES so that the almost linearly dependent vectors during the block Arnoldi process are kept and reintroduced in the next iterations. Calandra et al. [8] have suggested an extended version of the block restarted GMRES method, which utilizes the joint advantage of the flexible preconditioning technique and the deflation strategy to find the possible convergence of a linear combination of the systems. Later, Calandra et al. [7] have also proposed a new variant of the block flexible GMRES method which employs the deflation at each iteration and flexible preconditioning for solving non-Hermitian linear systems with multiple right-hand sides. In addition to these developments, Wu et al. [42] have proposed a residual-based simpler block GMRES method with deflation (RB-SBGMRES-D) in which a deflation procedure based on that shown by Calandra et al. [7] is used to identify the possible linear dependence of the residuals and the rank deficiency problem in the block Arnoldi process.

Furthermore, we can refer to the high ability of deflated versions of the GMRES and SGMRES methods to encounter the problem of multiple shifts and multiple right-hand sides systems. For instance, Sun et al. [39] have propounded a novel shifted BGMRES method in which the concept of the numerical rank of the block residual, developed by Robbé and Sadkane [32], has been exploited to tackle the inexact breakdowns in BGMRES. Thereupon, Sun et al. [40] have put forward the idea of incorporating the variable preconditioning into the vector deflation framework to create an effective shifted BGMRES method. Elbouyahyaoui et al. [13] have applied the property of shift-invariance associated with the block subspace to propose some frameworks, based on the eigenvalue deflation strategy suggested in [29], for the shifted BGMRES method. Besides these examples, Zhong and Gu [45] have offered a shifted variant of the adaptive SGMRES method which merges the deflated restarting strategy established in [5] and the variable preconditioning in an unified framework.

To the best of our knowledge, there is still no work that considers simultaneous eigenvalue deflation and vector deflation in the literature of the SBGMRES method. In this paper, by taking the merits of the vector deflation strategy proposed by Calandra et al. [7] and the idea of the eigenvalue deflation technique described by Meng et al. [29], we provide two new frameworks for the SBGMRES method to handle the linear systems of equations with multiple right-hand sides. To this end, we first apply the vector deflation technique to SBGMRES at each restart. Then, in order to speed up the convergence of SBGMRES with vector deflation, we propose a way that joins the eigenvalue deflation technique with the vector deflation process. These frameworks lead to the two specific implementations of SBGMRES, called the simpler block GMRES method with vector deflation restarting (SBGMRES-DR) and the deflated simpler block GMRES method with vector deflation restarting (D-SBGMRES-DR). In fact, the SBGMRES-DR method is proposed to remove linearly or almost linearly dependent vectors created by the block Arnoldi process. On the other hand, the D-SBGMRES-DR method applies the eigenvalue deflation and vector deflation simultaneously to the SBGMRES method. Several numerical experiments are performed to validate the performance of the proposed methods. Specifically, compared with some variants of BGMRES or SBGMRES methods based on the eigenvalue deflation or vector deflation, D-SBGMRES-DR is able to significantly reduce the matrix–vector products and CPU time.

This paper is organized as follows. In Sect. 2, the SBGMRES method is reviewed succinctly. The details of SBGMRES-DR are discussed in Sect. 3. The implementation of D-SBGMRES-DR is provided in Sect. 4. A theoretical result for SBGMRES-DR and D-SBGMRES-DR is shown in Sect. 5. Section 6 is devoted to the total computational cost of a cycle of D-SBGMRES-DR. The numerical results are given in Sect. 7. Finally, Sect. 8 draws a conclusion.

Throughout this paper, the notation $\|A\|_F$ denotes the Frobenius norm of a given matrix A. The notations I_k and $0_{i\times j}$ indicate the $k \times k$ identity matrix and the $i \times j$ zero matrix, respectively. The range space of the matrix A denoted by $\mathrm{range}(A)$ is the subspace generated by the columns of A. The singular values of the rectangular matrix $A \in \mathbb{C}^{m\times n}$ of rank l are denoted by $\sigma_1(A) \geq \sigma_2(A) \geq \cdots \geq \sigma_l(A) > 0$. The superscript H is used to indicate the transpose of a complex matrix A, which means that $A^H = \bar{A}^T$. Assume that $\mathcal{U}$ is a subspace of $\mathbb{C}^n$. The notation $X \perp \mathcal{U}$ means that $Y^H X = 0$ for all $Y \in \mathcal{U}$. For a matrix A, the MATLAB notations $A(1:i, 1:j)$, $A(:, j)$ and $A(i, j)$ stand for the first i rows and the first j columns of A, the j-th column of A and the (i, j)-th element of A, respectively. In addition, the block Krylov subspace $\mathcal{K}_m(A, V) := range\left[V, AV, \ldots, A^{m-1}V\right]$ is generated by the columns of the matrices $V, AV, \ldots, A^{m-1}V$ and is defined by

$$\mathcal{K}_m(A, V) = \left\{ \sum_{i=1}^{m} A^{i-1} V \Omega_i \;\middle|\; \Omega_i \in \mathbb{C}^{p\times p}, i = 1, 2, \ldots, m \right\}.$$

2 The SBGMRES Method

In this section, the SBGMRES method [27] is recalled briefly. Let $R_0 = B - AX_0$ be a block residual associated with the initial block guess $X_0 \in \mathbb{C}^{n\times p}$, and consider that $AR_0 = V_1 U_{11}$ is the reduced QR-factorization of AR_0 such that $V_1 \in \mathbb{C}^{n\times p}$ is a matrix with orthonormal columns and $U_{11} \in \mathbb{C}^{p\times p}$ is an upper triangular matrix of full rank. Moreover, assume that an orthonormal basis $\{V_1, V_2, \ldots, V_m\}$ for the subspace $A\mathcal{K}_m(A, R_0)$ is generated by applying the block Arnoldi algorithm. Hence, it can be seen that the relation $A[R_0, \mathcal{V}_{m-1}] = \mathcal{V}_m \mathcal{U}_m$ holds, where $\mathcal{V}_m = [V_1, V_2, \ldots, V_m] \in \mathbb{C}^{n\times mp}$ and $\mathcal{U}_m = [U_{ij}] \in \mathbb{C}^{mp\times mp}$ is an upper triangular matrix in which the blocks U_{ij} (for $i, j = 2, \ldots, m$) in the upper triangular part are constructed by the block Arnoldi algorithm.

At the m-th iteration of SBGMRES, the approximate solution X_m for (1) is sought such that

$$X_m = X_0 + [R_0, \mathcal{V}_{m-1}]Y_m \in X_0 + \mathcal{K}_m(A, R_0),$$

where the block vector $Y_m \in \mathbb{C}^{mp\times p}$ is determined by imposing the following condition

$$R_m = B - AX_m \perp A\mathcal{K}_m(A, R_0). \tag{2}$$

Now, thanks to the relation $R_m = R_0 - \mathcal{V}_m \mathcal{U}_m Y_m$, it follows from (2) that $\mathcal{U}_m Y_m = \mathcal{V}_m^H R_0$. Consequently, it can be checked that

$$R_m = (I - \mathcal{V}_m \mathcal{V}_m^H) R_0 = R_{m-1} - V_m F_m,$$

where $F_m = V_m^H R_{m-1}$.

The SBGMRES method is mentioned in Algorithm 1. Here, it is important to highlight that in Algorithms 1, $R_0/\|R_0\|_F$ is used instead of R_0 for the sake of stability.

Algorithm 1 The simpler block GMRES (SBGMRES) method [27].

Input: $A \in \mathbb{C}^{n\times n}$, $B \in \mathbb{C}^{n\times p}$, $X_0 \in \mathbb{C}^{n\times p}$, $tol > 0$, an integer $m > 0$.
Output: $X_j \in \mathbb{C}^{n\times p}$.
1: Compute $R_0 = B - AX_0$.
2: Compute the reduced QR-factorization of AR_0 as $AR_0 = V_1 U_{11}$.
3: **for** $j = 1, \ldots, m$ **do**
4: Compute $F_j = V_j^H R_{j-1}$ and $R_j = R_{j-1} - V_j F_j$.
5: **if** $\|R_j\|_F < tol$, **then** go to 12.
6: Set $\hat{V}_j = AV_j$.
7: **for** $i = 1, \ldots, j$ **do**
8: Set $U_{i,j+1} = V_i^H \hat{V}_j$ and $\hat{V}_j = \hat{V}_j - V_i U_{i,j+1}$.
9: **end for**
10: Compute the reduced QR-factorization $\hat{V}_j$ as $\hat{V}_j = V_{j+1} U_{j+1,j+1}$.
11: **end for**
12: Solve the upper triangular system: $\mathcal{U}_j Y_j = [F_1^T, \ldots, F_j^T]^T$.
13: Compute $X_j = X_0 + [R_0, \mathcal{V}_{j-1}] Y_j$.
14: **if** $\|R_j\|_F = \|B - AX_j\|_F < tol$, **then** stop.
15: Set $X_0 = X_j$ and $R_0 = R_j$, and then go to 2.

---

3 The SBGMRES Method with Vector Deflation Restarting

In this section, we propose a modified variant of the SBGMRES method for solving the linear systems of equations (1), which makes use of the vector deflation technique discussed by Calandra et al. [7]. To this end, the details of the deflated block Arnoldi process are reviewed at first. Then, the framework of the proposed SBGMRES method with vector deflation restarting (SBGMRES-DR) is established.

3.1 The Deflated Block Arnoldi Process

The explicit block size reduction, which is also known as the deflation strategy, is a formidable way to improve the convergence behavior of block GMRES-based methods [21]. When applying the block Arnoldi algorithm to generate an orthonormal basis for a block Krylov subspace, the possible linear dependency of the block right-hand side B or of the initial block residual R_0 can occur. A practicable remedy to deal with this situation is to explicitly reduce the number of right-hand sides with the aim of deleting useless information from the block Krylov subspace. In fact, the main concept behind the deflation strategy is to discover the linearly or almost linearly dependence vectors during the block iterative process and to deflate the vectors from the search subspace.

Robbé and Sadkane [32] have recently established a deflation strategy for the block Arnoldi algorithm, which is based on the numerical rank of the block residual and attempts to enhance the quality of the orthogonalization by ignoring linearly dependent vectors of the block Krylov basis and by possibly reintroducing them in the next block Arnoldi iteration.

To explain the deflated block Arnoldi process, let $R_0 \in \mathbb{C}^{n\times p}$ be an initial block residual matrix whose columns form the p Krylov directions. In practice, in order to detect a possible linear dependence relation between the columns of the block residual R_0, the deflated block Arnoldi process is applied such that $range(R_0)$ is decomposed as

$$range(R_0) = range(V_1) \oplus range(Q_1) \quad \text{with} \quad [V_1 \quad Q_1]^H [V_1 \quad Q_1] = I_p, \tag{3}$$

in which $V_1 \in \mathbb{C}^{n\times c_1}$, $Q_1 \in \mathbb{C}^{n\times r_1}$ and $c_1 + r_1 = p$.

Let us consider that the matrix $K \in \mathbb{C}^{n\times c_1}$ is an orthonormal matrix that includes all the c_1 Krylov directions at the $(j-1)$-th iteration. It should be noted that all the c_1 Krylov directions for the convergence of SBGMRES may not be required. For this reason, in order to decrease the computational costs, the deflated block Arnoldi process is applied in a way that the k_j Krylov directions are considered at the j-th iteration, while the d_j directions are deflated at the same iteration in which it is assumed that $k_j + d_j = c_1$. In fact, $range(K)$ is decomposed as follows

$$range(K) = range(W_j) \oplus range(P_j) \quad \text{with} \quad [W_j \quad P_j]^H [W_j \quad P_j] = I_{c_1}, \tag{4}$$

where $W_j \in \mathbb{C}^{n\times k_j}$ and $P_j \in \mathbb{C}^{n\times d_j}$. In light of this decomposition, it is worth mentioning that the use of k_j directions of W_j will have a significant impact on reducing the matrix–vector products [7].

In Algorithm 2, the deflated block Arnoldi process is summarized.

Algorithm 2 The deflated block Arnoldi process [7].

Input: $A \in \mathbb{C}^{n\times n}$, $[\mathcal{W}_{j-1} \quad P_{j-1}] \in \mathbb{C}^{n\times(s_{j-2}+c_1)}$ with $\mathcal{W}_{j-1} = [W_1, W_2, \ldots, W_{j-1}]$, $W_i \in \mathbb{C}^{n\times k_i}$ such that $W_i^H W_i = I_{k_i}$, $P_{j-1} \in \mathbb{C}^{n\times d_{j-1}}$, $k_{j-1} + d_{j-1} = c_1$, an integer $m > 0$.

Output: $\hat{\mathcal{W}}_j = [\mathcal{W}_{j-1} \quad P_{j-1} \quad \hat{W}_j] \in \mathbb{C}^{n\times(s_{j-1}+c_1)}$, $\begin{bmatrix} T_j \\ T_{jj} \end{bmatrix} \in \mathbb{C}^{(s_j+c_1)\times k_{j-1}}$.

1: Define $s_{j-2} = \sum_{i=1}^{j-2} k_i$ with $s_0 = 0$.
2: Compute $\hat{W} = AW_{j-1}$.
3: **for** $i = 1, \ldots, j-1$ **do**
4: Set $T_{i,j} = W_i^H \hat{W}$.
5: Set $\hat{W} = \hat{W} - W_i T_{i,j}$.
6: **end for**
7: Set $T_p = P_{j-1}^H \hat{W}$.
8: Set $\hat{W} = \hat{W} - P_{j-1} T_p$.
9: $T_j = \left[T_{1j}^H, \ldots, T_{j-1,j}^H, T_p^H\right]^H \in \mathbb{C}^{(s_{j-2}+c_1)\times k_{j-1}}$.
10: Compute the reduced QR-factorization of $\hat{W}$ as $\hat{W} = \hat{W}_j T_{jj}$, where $\hat{W}_j \in \mathbb{C}^{n\times k_{j-1}}$ and $T_{jj} \in \mathbb{C}^{k_{j-1}\times k_{j-1}}$.
11: Define s_{j-1} as $s_{j-1} = s_{j-2} + k_{j-1}$.
12: Define $\mathcal{W}_{j-1} \in \mathbb{C}^{n\times s_{j-1}}$ as $\mathcal{W}_{j-1} = [W_1, W_2, \ldots, W_{j-1}]$ and $\hat{\mathcal{W}}_{j-1} \in \mathbb{C}^{n\times(s_{j-1}+c_1)}$ as $\hat{\mathcal{W}}_{j-1} = [\mathcal{W}_{j-1} \quad P_{j-1} \quad \hat{W}_j]$ such that $AW_{j-1} = [\mathcal{W}_{j-1} \quad P_{j-1} \quad \hat{W}_j] \begin{bmatrix} T_j \\ T_{jj} \end{bmatrix}$.

3.2 The Framework of SBGMRES-DR

In this subsection, the framework of SBGMRES-DR for solving the linear systems of equations (1) is proposed. To do so, let $X_0 \in \mathbb{C}^{n\times p}$ be an initial guess and R_0 be its corresponding residual, and let

$$\mathcal{K}_m(A, R_0) = range(R_0) \oplus A\mathcal{K}_{m-1}(A, R_0) = range\{V_1, W_1, \ldots, W_{m-1}\},$$

where $\{V_1, W_1, \ldots, W_{m-1}\}$ is a basis for the Krylov subspace $\mathcal{K}_m(A, R_0)$. In the following of this subsection, the details of the derivation of V_1 and $\{W_1, W_2, \ldots, W_{m-1}\}$ are provided.

3.2.1 Construction of V_1

Let us consider that the reduced QR-factorization of R_0 is of the form $R_0 = \hat{\mathcal{V}}_1\hat{\Omega}_0$, where $\hat{\mathcal{V}}_1 \in \mathbb{C}^{n\times p}$ is an orthonormal matrix and $\hat{\Omega}_0 \in \mathbb{C}^{p\times p}$ is an upper triangular matrix. We also assume that the SVD of the matrix $\hat{\Omega}_0$ is expressed by $\hat{\Omega}_0 = U\Sigma W^H$. As presented in [8,32], the small singular values of $\hat{\Omega}_0$ play an important role to approximate the linear dependency of the residual columns. A common approach for the detection of the c_1 (where $c_1 \leq p$) largest singular values of $\hat{\Omega}_0$ is to filter the singular values of $\hat{\Omega}_0$ based on the following deflation condition, i.e.,

$$\sigma_i(\hat{\Omega}_0) > \epsilon_d tol, \quad 1 \leq i \leq c_1, \tag{5}$$

where ϵ_d is a positive deflation threshold, and tol indicates the convergence threshold. Consequently, using the relation (5), the matrix Σ can be decomposed as:

$$\Sigma = \begin{bmatrix} \Sigma^+ & 0_{c_1\times(p-c_1)} \\ 0_{(p-c_1)\times c_1} & \Sigma^- \end{bmatrix},$$

in which the matrices $\Sigma^+ \in \mathbb{C}^{c_1\times c_1}$ and $\Sigma^- \in \mathbb{C}^{r_1\times r_1}$, where $r_1 = p - c_1$, are defined as $\Sigma^+ = \Sigma(1 : c_1, 1 : c_1)$ and $\Sigma^- = \Sigma(c_1 + 1 : p, c_1 + 1 : p)$, respectively. According to the deflation condition (5), it can be observed that

$$\|\Sigma^+\|_2 > \epsilon_d\, tol \quad \text{and} \quad \|\Sigma^-\|_2 \leq \epsilon_d\, tol. \tag{6}$$

Meanwhile, if it is assumed that the matrices $U^+ \in \mathbb{C}^{p\times c_1}$, $U^- \in \mathbb{C}^{p\times r_1}$, $W^+ \in \mathbb{C}^{p\times c_1}$ and $W^- \in \mathbb{C}^{p\times r_1}$ are defined as $U^+ = U(:, 1 : c_1)$, $U^- = U(:, c_1 + 1 : p)$, $W^+ = W(:, 1 : c_1)$ and $W^- = W(:, c_1 + 1 : p)$, respectively, then the initial block residual R_0 can be written as

$$R_0 = \hat{\mathcal{V}}_1 \begin{bmatrix} U^+ & U^- \end{bmatrix} \begin{bmatrix} \Sigma^+ & 0 \\ 0 & \Sigma^- \end{bmatrix} \begin{bmatrix} W^+ & W^- \end{bmatrix} = \hat{\mathcal{V}}_1 U^+\Sigma^+ {W^+}^H + \hat{\mathcal{V}}_1 U^-\Sigma^- {W^-}^H. \tag{7}$$

In the following, we consider two cases for c_1: the first case is $c_1 < p$, and the second case is $c_1 = p$.

Case 1. In this case, since $c_1 < p$ and $r_1 = p - c_1$, it can infer that there exist r_1 linearly or almost linearly dependent vectors in the initial block residual. Let us assume that the linearly independent vectors and linearly or almost linearly dependent vectors are stored in the matrices V_1 and Q_1, respectively. In the following, we explain how the matrices V_1 and Q_1 can be constructed. Since W is a unitary matrix, it can be seen from (5) and (7) that

$$\|R_0 W^+\|_2 > \epsilon_d\, tol, \quad \text{and} \quad \|R_0 W^-\|_2 \leq \epsilon_d\, tol.$$

In view of the above relations, it turns out that $\sigma_i(R_0W^+) > \epsilon_d tol$, for $i = 1, 2, \ldots, c_1$, and $\sigma_i(R_0W^-) \le \epsilon_d tol$, for $i = 1, 2, \ldots, r_1$. In fact, $range(R_0W^+)$ and $range(R_0W^-)$ contain the linearly independent and linearly dependent or almost linearly dependent vectors, respectively. By setting

$$range(V_1) = range(\hat{\mathcal{V}}_1U^+) = range(\hat{\mathcal{V}}_1U^+\Sigma^+) = range(\hat{\mathcal{V}}_1\hat{\Omega}_0W^+) = range(R_0W^+),$$
$$range(Q_1) = range(\hat{\mathcal{V}}_1U^-) = range(\hat{\mathcal{V}}_1U^-\Sigma^-) = range(\hat{\mathcal{V}}_1\hat{\Omega}_0W^-) = range(R_0W^-),$$

it can be verified that

$$range[V_1 \quad Q_1] = range(R_0W) = range(\hat{\mathcal{V}}_1\hat{\Omega}_0W). \tag{8}$$

Let $\hat{\Omega}_0W = E_1\tilde{\Omega}_0$ be the reduced QR-factorization of $\hat{\Omega}_0W$, where $E_1 \in \mathbb{C}^{p\times p}$ and $\tilde{\Omega}_0 \in \mathbb{C}^{p\times p}$. This factorization allows us to define the first block vector basis of the deflated block Arnoldi process as follows

$$\hat{\mathcal{V}}_1E_1 = \begin{bmatrix} V_1 & Q_1 \end{bmatrix}, \tag{9}$$

where $V_1 \in \mathbb{C}^{n\times c_1}$ and $Q_1 \in \mathbb{C}^{n\times r_1}$.

Case 2. In this case, by the assumption that $c_1 = p$ (which is equivalent to say that $r_1 = 0$), it follows immediately that $Q_1 = 0$ and all of the vectors in the initial block residual are linearly independent. In this case, it is also assumed that $V_1 = \hat{\mathcal{V}}_1$.

Based on the above discussion, we can see that the initial block residual R_0 can be expressed as

$$R_0 = \hat{\mathcal{V}}_1\hat{\Omega}_0 = \hat{\mathcal{V}}_1E_1E_1^H\hat{\Omega}_0 = \begin{bmatrix} V_1 & Q_1 \end{bmatrix}\Omega_0,$$

where $\Omega_0 = E_1^H\hat{\Omega}_0$.

3.2.2 Construction of $\{W_1, W_2, \ldots, W_{m-1}\}$

To construct $\{W_1, W_2, \ldots, W_{m-1}\}$, which is an orthonormal basis for $A\mathcal{K}_{m-1}(A, R_0)$, it is necessary to calculate the reduced QR-factorization of AV_1, i.e.,

$$AV_1 = \hat{\mathcal{W}}_1\hat{\Lambda}_0, \tag{10}$$

with $\hat{\mathcal{W}}_1 \in \mathbb{C}^{n\times c_1}$ and $\hat{\Lambda}_0 \in \mathbb{C}^{c_1\times c_1}$. Now, let us assume that the singular value decomposition of $\hat{\Lambda}_0$ is as follows

$$\hat{\Lambda}_0 = U\Sigma W^H, \tag{11}$$

where $U \in \mathbb{R}^{c_1\times c_1}$, $W \in \mathbb{R}^{c_1\times c_1}$ are orthonormal matrices and $\Sigma \in \mathbb{R}^{c_1\times c_1}$ is a diagonal matrix. Now, similar to that presented for the initial block residual R_0, our goal is to select some of the information that is relevant to the singular value decomposition mentioned in (11). To this end, by using the deflation condition (5), the numbers k_1 and d_1 are determined in a way that $k_1 + d_1 = c_1$. Therefore, AV_1 can be written as

$$AV_1 = \hat{\mathcal{W}}_1\hat{\Lambda}_0 = \hat{\mathcal{W}}_1U^+\Sigma^+{W^+}^H + \hat{\mathcal{W}}_1U^-\Sigma^-{W^-}^H,$$

where $U^+, W^+ \in \mathbb{C}^{c_1\times k_1}$, $U^-, W^- \in \mathbb{C}^{c_1\times k_2}$, $\Sigma^+ \in \mathbb{C}^{k_1\times k_1}$ and $\Sigma^- \in \mathbb{C}^{k_2\times k_2}$.

Moreover, similar to Case 1 mentioned in Sect. 3.2.1, the search space spanned by the linearly independent vectors ($range(W_1)$) and linearly dependent vectors ($range(P_1)$) should satisfy the following relations

$$range(W_1) = range(\hat{\mathcal{W}}_1U^+) = range(\hat{\mathcal{W}}_1U^+\Sigma^+) = range(\hat{\mathcal{W}}_1\hat{\Lambda}_0W^+), \tag{12}$$

$$range(P_1) = range(\hat{\mathcal{W}}_1 U^-) = range(\hat{\mathcal{W}}_1 U^- \Sigma^-) = range(\hat{\mathcal{W}}_1 \hat{\Lambda}_0 W^-). \quad (13)$$

Therefore, by combining the two relations (12) and (12), we get

$$range([W_1 \quad P_1]) = range(\hat{\mathcal{W}}_1 \hat{\Lambda}_0 W) = range(\hat{\mathcal{W}}_1 G_1 \Lambda_0), \quad (14)$$

where $G_1 \Lambda_0$ is the QR-factorization of $\hat{\Lambda}_0 W$ with $G_1, \Lambda_0 \in \mathbb{C}^{c_1 \times c_1}$. In view of (14), we choose W_1 and P_1 as the first k_1 and the last d_1 columns of $\hat{\mathcal{W}}_1 G_1 = [W_1 \quad P_1]$. Hence, we can rewrite the relation (10) as the following deflated block Arnoldi relation

$$AV_1 = \hat{\mathcal{W}}_1 G_1 G_1^H \hat{\Lambda}_0 = [W_1 \quad P_1] \mathcal{T}_1,$$

where $\mathcal{T}_1 = G_1^H \hat{\Lambda}_0 \in \mathbb{C}^{c_1 \times c_1}$.

Now, suppose that the $(j-1)$ Krylov block vectors are constructed and also assume that the following relation holds at the begining of the j-th iteration of the deflated block Arnoldi relation, i.e.,

$$A\Big[V_1 \quad \mathcal{W}_{j-2}\Big] = [\mathcal{W}_{j-1} \quad P_{j-1}] \mathcal{T}_{j-1}, \quad j > 2, \quad (15)$$

where $\mathcal{W}_{j-2} \in \mathbb{C}^{n \times s_{j-2}}$, $\mathcal{W}_{j-1} \in \mathbb{C}^{n \times s_{j-1}}$, $P_{j-1} \in \mathbb{C}^{n \times d_{j-1}}$, $\mathcal{T}_{j-1} \in \mathbb{C}^{(s_{j-2}+c_1) \times (s_{j-2}+c_1)}$ and $s_{j-1} = s_{j-2} + k_{j-1}$, $s_0 = 0$. Note that the columns of the matrix $[\mathcal{W}_{j-1} \quad P_{j-1}]$ are orthonormal, that is,

$$[\mathcal{W}_{j-1} \quad P_{j-1}]^H [\mathcal{W}_{j-1} \quad P_{j-1}] = I_{(s_{j-2}+c_1)}.$$

In order to produce the matrices $\hat{W}_j \in \mathbb{C}^{n \times k_{j-1}}$ and $\hat{\mathcal{T}}_j \in \mathbb{C}^{(s_{j-1}+c_1) \times (s_{j-1}+c_1)}$, we need to perform the j-th iteration of the deflated block Arnoldi process satisfying

$$A\Big[V_1 \quad \mathcal{W}_{j-2} \quad W_{j-1}\Big] = \Big[[\mathcal{W}_{j-1} \quad P_{j-1}] \quad \hat{W}_j\Big] \hat{\mathcal{T}}_j, \quad (16)$$

where

$$\hat{\mathcal{T}}_j = \begin{bmatrix} \mathcal{T}_{j-1} & T_j \\ 0_{k_{j-1} \times (s_{j-2}+c_1)} & T_{jj} \end{bmatrix} \in \mathbb{C}^{(s_{j-1}+c_1) \times (s_{j-1}+c_1)},$$

with $T_j \in \mathbb{C}^{(s_{j-2}+c_1) \times k_{j-1}}$ and $T_{jj} \in \mathbb{C}^{k_{j-1} \times k_{j-1}}$.

In the next step, we need to consider the subspace decomposition mentioned in (4). To achieve this goal, by applying the idea suggested in [7], we find a unitary matrix $\mathcal{G}_j \in \mathbb{C}^{(s_{j-1}+c_1) \times (s_{j-1}+c_1)}$ in a way that

$$\Big[[\mathcal{W}_{j-1} \quad P_{j-1}] \quad \hat{W}_j\Big] \mathcal{G}_j = \Big[\mathcal{W}_{j-1} \quad W_j \quad P_j\Big]. \quad (17)$$

Since $\mathcal{W}_{j-1}^H \Big[P_{j-1} \quad \hat{W}_{j-1}\Big] = 0_{s_{j-1} \times c_1}$, we conclude that the matrix form of $\mathcal{G}_j$ is as follows:

$$\mathcal{G}_j = \begin{bmatrix} I_{s_{j-1}} & 0_{s_{j-1} \times c_1} \\ 0_{c_1 \times s_{j-1}} & G_j \end{bmatrix}, \quad (18)$$

where $G_j^H G_j = I_{c_1}$. Now, by combining the relations (16) and (17), we obtain a deflated block Arnoldi relation as follows:

$$A\Big[V_1 \quad \mathcal{W}_{j-1}\Big] = \Big[[\mathcal{W}_{j-1} \quad P_{j-1}] \quad \hat{W}_j\Big] \mathcal{G}_j \mathcal{G}_j^H \hat{\mathcal{T}}_j = \Big[\mathcal{W}_j \quad P_j\Big] \mathcal{T}_j. \quad (19)$$

In the sequel, we consider how to build the matrix $\mathcal{G}_j$. First, we assume that the matrices $\mathcal{T}_{j-1}$ and T_j are partitioned as

$$\mathcal{T}_{j-1} = \begin{bmatrix} \mathcal{T}_{j-1}{}^+ \\ \mathcal{T}_{j-1}{}^- \end{bmatrix} \quad \text{and} \quad T_j = \begin{bmatrix} T_j{}^+ \\ T_j{}^- \end{bmatrix},$$

where $\mathcal{T}_{j-1}{}^{+} = \mathcal{T}_{j-1}(1 : s_{j-1}, 1 : s_{j-2} + c_1)$, $\mathcal{T}_{j-1}{}^{-} = \mathcal{T}_{j-1}(s_{j-1} + 1 : s_{j-1} + d_{j-1}, 1 : s_{j-2} + c_1)$, $T_j{}^{+} = T_j(1 : s_{j-1}, 1 : k_{j-1})$ and $T_j{}^{-} = T_j(s_{j-1} + 1 : s_{j-1} + d_{j-1}, 1 : k_{j-1})$. Then, we define

$$\hat{\Lambda}_{j-1} = \begin{bmatrix} \mathcal{T}_{j-1}{}^{-} & T_j{}^{-} \\ 0_{k_{j-1}\times(s_{j-2}+c_1)} & T_{jj} \end{bmatrix} \in \mathbb{C}^{c_1\times(s_{j-1}+c_1)}.$$

To determine the unitary matrix $G_j \in \mathbb{C}^{c_1\times c_1}$ and the numbers k_j, d_j such that $k_j + d_j = c_1$, we compute the SVD of $\hat{\Lambda}_{j-1}$ as $\hat{\Lambda}_{j-1} = U\Sigma W^H$, where $U \in \mathbb{C}^{c_1\times c_1}$, $W \in \mathbb{C}^{(s_{j-1}+c_1)\times(s_{j-1}+c_1)}$ are orthonormal matrices and $\Sigma \in \mathbb{C}^{c_1\times(s_{j-1}+c_1)}$ is a diagonal matrix. Now, by considering the condition (4), we choose $range(W_j)$ and $range(P_j)$ in such a way that the orthonormal relation $W_j \perp \mathcal{W}_{j-1}$ and the following relations are satisfied:

$$\begin{aligned} range(W_j) &= range\left((I - \mathcal{W}_{j-1}\mathcal{W}_{j-1}^H)[\mathcal{W}_{j-1} \quad P_{j-1} \quad \hat{W}_j]\begin{bmatrix} 0_{s_{j-1}\times k_j} \\ U^{+} \end{bmatrix}\right) \\ &= range\left([0_{n\times s_{j-1}} \quad P_{j-1} \quad \hat{W}_j]\begin{bmatrix} 0_{s_{j-1}\times k_j} \\ U^{+} \end{bmatrix}\right) \\ &= range\left([P_{j-1} \quad \hat{W}_j]U^{+}\Sigma^{+}\right) = range\left([P_{j-1} \quad \hat{W}_j]\hat{\Lambda}_{j-1}W^{+}\right), \end{aligned} \tag{20}$$

and

$$\begin{aligned} range(P_j) &= range\left((I - \mathcal{W}_{j-1}\mathcal{W}_{j-1}^H)[\mathcal{W}_{j-1} \quad P_{j-1} \quad \hat{W}_j]\begin{bmatrix} 0_{s_{j-1}\times k_j} \\ U^{-} \end{bmatrix}\right) \\ &= range\left([0_{n\times s_{j-1}} \quad P_{j-1} \quad \hat{W}_j]\begin{bmatrix} 0_{s_{j-1}\times k_j} \\ U^{-} \end{bmatrix}\right) \\ &= range\left([P_{j-1} \quad \hat{W}_j]U^{-}\Sigma^{-}) = range([P_{j-1} \quad \hat{W}_j]\hat{\Lambda}_{j-1}W^{-}\right). \end{aligned} \tag{21}$$

Therefore, by using the relations (20) and (21), we have

$$range\left([W_j \quad P_j]) = range([P_{j-1} \quad \hat{W}_j]\hat{\Lambda}_{j-1}W\right), \tag{22}$$

in which $W_j \in \mathbb{C}^{n\times k_j}$ and $P_j \in \mathbb{C}^{n\times d_j}$. Now, according to the relation (22), the unitary matrix $G_j \in \mathbb{C}^{c_1\times c_1}$ can be obtained from the reduced QR-factorization of the matrix $\hat{\Lambda}_{j-1}W$, i.e., $\hat{\Lambda}_{j-1}W = G_j\Lambda_{j-1}$. In addition to this, it can be concluded that $[W_j \quad P_j] = [P_{j-1} \quad \hat{W}_j]G_j$.

The derivation of the matrix G_j and the parameters k_j and d_j is shown in Algorithm 3.

Algorithm 3 Determination of k_j, d_j, G_j and $\mathcal{G}_j$.

Input: Choose a deflation threshold ϵ_d and a threshold $tol > 0$.
Output: k_j, d_j, G_j and $\mathcal{G}_j$.

1: Compute the SVD of $\hat{\Lambda}_{j-1}$ as $\hat{\Lambda}_{j-1} = U\Sigma W^H$, where $U \in \mathbb{C}^{c_1\times c_1}$, $\Sigma \in \mathbb{C}^{c_1\times(s_{j-1}+c_1)}$ and $W \in \mathbb{C}^{(s_{j-1}+c_1)\times(s_{j-1}+c_1)}$.
2: Select the k_j singular values of $\hat{\Lambda}_{j-1}$ such that $\sigma_i(\hat{\Lambda}_{j-1}) > \epsilon_d tol$ for $1 \le i \le k_j$.
3: Set $d_j = c_1 - k_j$.
4: Compute the reduced QR-factorization of $\hat{\Lambda}_{j-1}W$ as $\hat{\Lambda}_{j-1}W = G_j\Lambda_{j-1}$, where $G_j \in \mathbb{C}^{c_1\times c_1}$, $\Lambda_{j-1} \in \mathbb{C}^{c_1\times c_1}$ and $G_j^H G_j = I_{c_1}$.
5: Define $\mathcal{G}_j \in \mathbb{C}^{(s_{j-1}+c_1)\times(s_{j-1}+c_1)}$ as $\mathcal{G}_j = \begin{pmatrix} I_{s_{j-1}} & 0_{s_{j-1}\times c_1} \\ 0_{c_1\times s_{j-1}} & G_j \end{pmatrix}$.

3.3 Algorithm of SBGMRES-DR

According to the explanations and analyzes provided in the previous section, the SBGMRES-DR method can be described in Algorithm 4. It should be also noted that before showing this algorithm, we present a practical form for the approximate solution X_j and its corresponding residual R_j.

Theorem 1 *Suppose that the set of matrices* $\{W_1, \ldots, W_j, P_j\}$ *and the block upper triangular matrix* $\mathcal{T}_j$ *are obtained at the end of a cycle of SBGMRES-DR. Then, the approximate solution* X_j *and the corresponding residual* R_j *are presented as*

$$\begin{aligned} X_j &= X_0 + [V_1 \quad \mathcal{W}_{j-1}]Y_j, \\ R_j &= R_{j-1} - W_j S_j, \end{aligned}$$

where $\mathcal{W}_{j-1} = [W_1, W_2, \ldots, W_{j-1}]$, $Y_j = \mathcal{T}_j^{-1}[S_1^H, \ldots, S_j^H, P_j^H R_{j-1}]^H$ *and* $S_j = W_j^H R_{j-1}$.

Proof Let X_0 be an initial guess with the corresponding residual R_0. At the j-th step of SBGMRES-DR, the approximate solution X_j can be expressed as

$$X_j = X_0 + [V_1 \quad \mathcal{W}_{j-1}]Y_j,$$

where $Y_j \in \mathbb{C}^{(s_{j-1}+c_1)\times p}$. The residual R_j associated with X_j also satisfies the following orthogonality condition

$$R_j = R_0 - A[V_1 \quad \mathcal{W}_{j-1}]Y_j \perp A\mathcal{K}_j(A, R_0).$$

Using this orthogonality condition and the relation (19) together with the fact that

$$[\mathcal{W}_j \quad P_j]^H[\mathcal{W}_j \quad P_j] = I_{(s_{j-1}+c_1)},$$

implies that $\mathcal{T}_j Y_j = [\mathcal{W}_j \quad P_j]^H R_0$. Let us define $S_i = W_i^H R_0$, for $i = 1, \ldots, j$. Then, we have

$$[\mathcal{W}_j \quad P_j]^H R_0 = [S_1^H, \ldots, S_j^H, P_j^H R_{j-1}]^H.$$

With this relation, it follows that Y_j can be determined by solving the following triangular system

$$\mathcal{T}_j Y_j = [S_1^H, \ldots, S_j^H, P_j^H R_{j-1}]^H.$$

As a result, the residual R_j is of the form

$$\begin{aligned} R_j &= R_0 - [\mathcal{W}_j \quad P_j]\mathcal{T}_j Y_j \\ &= R_0 - [\mathcal{W}_j \quad P_j][\mathcal{W}_j \quad P_j]^H R_0 \\ &= R_0 - \mathcal{W}_{j-1}\mathcal{W}_{j-1}^H R_0 - W_j W_j^H R_0 \\ &= R_{j-1} - W_j W_j^H R_0. \end{aligned}$$

Moreover, from the orthogonality condition $R_j \perp A\mathcal{K}_j(A, R_0)$, it can be shown that $S_j = W_j^H R_{j-1}$. Therefore, the residual R_j is displayed as $R_j = R_{j-1} - W_j S_j$. □

The entire process of the SBGMRES-DR method for solving (1) is outlined in Algorithm 4.

Algorithm 4 The SBGMRES method with vector deflation restarting (SBGMRES-DR).

Input: $A \in \mathbb{C}^{n\times n}$, $B \in \mathbb{C}^{n\times p}$. Choose a threshold tol, a deflation threshold ϵ_d, the maximum number of restart cycles N and an initial guess $X_0 \in \mathbb{C}^{n\times p}$.
Output: $X_m \in \mathbb{C}^{n\times p}$.
1: Compute the initial block residual: $R_0 = B - AX_0$.
2: **for** cycle $= 1, \ldots, N$ **do**
3: Compute the reduced QR-factorization of R_0 as $R_0 = \hat{\mathcal{V}}_1 \hat{\Omega}_0$ with $\hat{\mathcal{V}}_1 \in \mathbb{C}^{n\times p}$ and $\hat{\Omega}_0 \in \mathbb{C}^{p\times p}$.
4: Compute the SVD of $\hat{\Omega}_0$ as $\hat{\Omega}_0 = U\Sigma W^H$.
5: Select c_1 singular values of $\hat{\Omega}_0$ such that $\sigma_i(\hat{\Omega}_0) > \epsilon_d tol$ for all i such that $1 \le i \le c_1$.
6: Compute the reduced QR-factorization of $\hat{\Omega}_0 W$ as $\hat{\Omega}_0 W = E_1 \tilde{\Omega}_0$, where $E_1, \tilde{\Omega}_0 \in \mathbb{C}^{p\times p}$.
7: Define $V_1 \in \mathbb{C}^{n\times c_1}$ as $V_1 = \hat{\mathcal{V}}_1 E_1(:, 1 : c_1)$.
8: Compute the reduced QR-factorization of AV_1 as $AV_1 = \hat{\mathcal{W}}_1 \hat{\Lambda}_0$ with $\hat{\mathcal{W}}_1 \in \mathbb{C}^{n\times c_1}$ and $\hat{\Lambda}_0 \in \mathbb{C}^{c_1\times c_1}$.
9: Use Algorithm 3 to determine the deflation unitary matrix $G_1 \in \mathbb{C}^{c_1\times c_1}$, and the numbers k_1 and d_1 such that $k_1 + d_1 = c_1$. Set $s_1 = k_1$.
10: Define $[\mathcal{W}_1 \quad P_1] = \hat{\mathcal{W}}_1 G_1$ with $\mathcal{W}_1 \in \mathbb{C}^{n\times s_1}$ as the first s_1 columns and $P_1 \in \mathbb{C}^{n\times d_1}$ as the last d_1 columns of $\hat{\mathcal{W}}_1 G_1$, respectively.
11: Define $W_1 = \mathcal{W}_1$ and $\mathcal{T}_1 = G_1^H \hat{\Lambda}_0 \in \mathbb{C}^{c_1\times c_1}$.
12: Compute $R_1 = R_0 - W_1 S_1$, $S_1 = W_1^H R_0$.
13: **if** $\|R_1\|_F < tol$, **then** go to 23.
14: **for** $j = 2, \ldots, m$ **do**
15: Apply Algorithm 2 to obtain $\hat{\mathcal{W}}_j \in \mathbb{C}^{n\times(s_{j-1}+c_1)}$, and the matrix $\hat{\mathcal{T}}_j \in \mathbb{C}^{(s_{j-1}+c_1)\times(s_{j-1}+c_1)}$ such that $A[V_1 \quad \mathcal{W}_{j-1}] = \hat{\mathcal{W}}_j \hat{\mathcal{T}}_j$ and $\hat{\mathcal{W}}_j = [\mathcal{W}_{j-1} \quad P_{j-1} \quad \hat{W}_j]$.
16: Set $\hat{\Lambda}_{j-1} = \begin{bmatrix} \mathcal{T}_{j-1}^{-} & T_j^{-} \\ 0_{k_{j-1}\times(s_{j-2}+c_1)} & T_{jj} \end{bmatrix} \in \mathbb{C}^{c_1\times(s_{j-1}+c_1)}$.
17: Use Algorithm 3 to determine the deflation unitary matrix $G_j \in \mathbb{C}^{c_1\times c_1}$, and the numbers k_j, d_j such that $k_j + d_j = c_1$. Set $s_j = s_{j-1} + k_j$.
18: Define $[W_j \quad P_j] = [P_{j-1} \quad \hat{W}_j]G_j$ with $W_j \in \mathbb{C}^{n\times k_j}$ and $P_j \in \mathbb{C}^{n\times d_j}$ as the first k_j and the last d_j columns of $[P_{j-1} \quad \hat{W}_j]G_j$, respectively.
19: Define $\mathcal{T}_j = \begin{bmatrix} \mathcal{T}_{j-1}^{+} \quad T_j^{+} \\ G_j^H \hat{\Lambda}_{j-1} \end{bmatrix} \in \mathbb{C}^{(s_{j-1}+c_1)\times(s_{j-1}+c_1)}$.
20: Compute $S_j = W_j^H R_{j-1}$ and $R_j = R_{j-1} - W_j S_j$.
21: **if** $\|R_j\|_F < tol$, **then** go to 23.
22: **end for**
23: Solve the following upper triangular system:

$$\mathcal{T}_j Y_j = [S_1^H, \ldots, S_j^H, (P_j^H R_0)^H]^H.$$

24: Compute $X_j = X_0 + [V_1 \quad \mathcal{W}_{j-1}]Y_j$ and $R_j = B - AX_j$.
25: **if** $\|R_j\|_F < tol$, **then** stop.
26: Set $X_0 = X_j$ and $R_0 = R_j$.
27: **end for**

4 The Deflated SBGMRES Method with Vector Deflation Restarting

In this section, we introduce the deflated simpler block GMRES method with vector deflation restarting (D-SBGMRES-DR) for solving the linear systems with multiple right-hand sides (1). This method is constructed from a combination of SBGMRES-DR with the eigenvalue deflation technique suggested by Meng et al. [29]. In the following, we first recall the definition of a harmonic Ritz pair, which plays a key role in the eigenvalue deflation technique. Then, the harmonic Ritz pairs associated with the basis obtained from a cycle of SBGMRES-DR are characterized. Finally, the structure of the block Arnoldi relation and its deflated version is analyzed.

Definition 1 [18] (*Harmonic Ritz pair*). Consider a subspace $\mathcal{U}$ of $\mathbb{C}^n$. Given a matrix $B \in \mathbb{C}^{n\times n}$, $\lambda \in \mathbb{C}$ and $y \in \mathcal{U}$, (λ, y) is a harmonic Ritz pair of B with respect to $\mathcal{U}$ if and only if

$$By - \lambda y \perp B\mathcal{U},$$

or equivalently, for the canonical scalar product,

$$\forall w \in range(B\mathcal{U}), \quad w^H(By - \lambda y) = 0.$$

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

In the D-SBGMRES-DR method, the eigenvectors associated with the small eigenvalues in magnitude are calculated at the end of a cycle of the block Arnoldi algorithm. In specific, the obtained spectral information at a restart can be recovered with the aim of reducing the computational costs of D-SBGMRES-DR. The following proposition shows how the harmonic Ritz pairs used in D-SBGMRES-DR can be computed.

Proposition 1 *Let $\mathcal{U} = span\{\mathcal{Z}_m\}$, where $\mathcal{Z}_m = [V_1 \quad \mathcal{W}_{m-1}]$ is built by SBGMRES-DR at the end of a cycle. The harmonic Ritz pairs $(\tilde{\theta}_i, \tilde{g}_i)$ corresponding to $\mathcal{Z}_m$ satisfy the following relations*

$$\hat{\mathcal{T}}_m^H \hat{\mathcal{T}}_m \tilde{g}_i - \tilde{\theta}_i \hat{\mathcal{T}}_m^H \begin{pmatrix} \mathcal{W}_{m-1}^H \mathcal{Z}_m \\ P_{m-1}^H \mathcal{Z}_m \\ \hat{W}_m^H \mathcal{Z}_m \end{pmatrix} \tilde{g}_i = 0, \quad i = 1, 2, \ldots, k + c_1, \tag{23}$$

where $\hat{\mathcal{T}}_m \in \mathbb{C}^{(s_{m-1}+c_1)\times(s_{m-1}+c_1)}$, $\mathcal{W}_{m-1} \in \mathbb{C}^{n\times s_{m-1}}$, $P_{m-1} \in \mathbb{C}^{n\times d_{m-1}}$, $\hat{W}_m \in \mathbb{R}^{n\times k_{m-1}}$ and $\mathcal{Z}_m \in \mathbb{C}^{n\times(s_{m-1}+c_1)}$ are defined by (16). Moreover, $\tilde{g}_i \in \mathbb{C}^{(s_{m-1}+c_1)}$ and $\mathcal{Z}_m \tilde{g}_i$ are the harmonic Ritz vectors corresponding to harmonic Ritz values $\tilde{\theta}_i$, for $i = 1, 2, \ldots, k + c_1$.

Proof Suppose that a first cycle of SBGMRES-DR is carried out. The use of Definition 1 indicates that the harmonic Ritz pairs satisfy the following orthogonality conditions:

$$A\mathcal{Z}_m \tilde{g}_i - \tilde{\theta}_i \mathcal{Z}_m \tilde{g}_i \perp range(A\mathcal{Z}_m), \tag{24}$$

for $i = 1, 2, \ldots, k + c_1$. Substituting $A\mathcal{Z}_m = [\mathcal{W}_{m-1} \quad P_{m-1} \quad \hat{W}_m]\hat{\mathcal{T}}_m$ into (24), we get

$$([\mathcal{W}_{m-1} \quad P_{m-1} \quad \hat{W}_m]\hat{\mathcal{T}}_m)^H([\mathcal{W}_{m-1} \quad P_{m-1} \quad \hat{W}_m]\hat{\mathcal{T}}_m \tilde{g}_i - \tilde{\theta}_i \mathcal{Z}_m \tilde{g}_i) = 0.$$

Since $[\mathcal{W}_{m-1} \quad P_{m-1} \quad \hat{W}_m]^H[\mathcal{W}_{m-1} \quad P_{m-1} \quad \hat{W}_m] = I_{s_{m-1}+c_1}$, the relations (23) are obtained. □

Proposition 1 reveals that if $\hat{\mathcal{T}}_m$ is nonsingular, then the k harmonic Ritz pairs can be computed through the following generalized eigenvalue problem

$$\hat{\mathcal{T}}_m \tilde{g}_i = \tilde{\theta}_i \begin{pmatrix} \mathcal{W}_{m-1}^H \mathcal{Z}_m \\ P_{m-1}^H \mathcal{Z}_m \\ \hat{W}_m \mathcal{Z}_m \end{pmatrix} \tilde{g}_i, \quad i = 1, 2, \ldots, k + c_1. \tag{25}$$

4.1 Discussion on the Deflated Block Arnoldi Relation

In this subsection, we will demonstrate that the block Arnoldi relation

$$A\mathcal{Z}_j = [\mathcal{W}_{j-1} \quad P_{j-1} \quad \hat{W}_j]\hat{\mathcal{T}}_j,$$

where $\mathcal{Z}_j = [V_1 \quad \mathcal{W}_{j-1}]$, still holds after constructing the initial block orthonormal basis vectors of the new search subspace. To begin with, let k be an integer number, and let

$\tilde{g}_1, \tilde{g}_2, \ldots, \tilde{g}_l$ be the harmonic Ritz vectors associated with the l (where $l = k + c_1$) smallest harmonic Ritz values of the generalized eigenpair problem (23) or (25). By defining $\tilde{\mathcal{G}}_l = [\tilde{g}_1, \ldots, \tilde{g}_l] \in \mathbb{C}^{(s_{m-1}+c_1)\times l}$, we compute the reduced QR-factorization of $\tilde{\mathcal{G}}_l$ as $\tilde{\mathcal{G}}_l = \tilde{\mathcal{Q}}_l \tilde{\mathcal{R}}_l$, where $\tilde{\mathcal{Q}}_l \in \mathbb{C}^{(s_{m-1}+c_1)\times l}$ has orthonormal columns and $\tilde{\mathcal{R}}_l \in \mathbb{C}^{l\times l}$ is a nonsingular upper triangular matrix. Multiplying $A\mathcal{Z}_m = [\mathcal{W}_{m-1} \quad P_{m-1} \quad \hat{W}_m]\hat{\mathcal{T}}_m$ by $\tilde{\mathcal{Q}}_l$ from the right, the block Arnoldi relation can be written as

$$A\mathcal{Z}_m \tilde{\mathcal{Q}}_l = [\mathcal{W}_{m-1} \quad P_{m-1} \quad \hat{W}_m]\hat{\mathcal{T}}_m \tilde{\mathcal{Q}}_l. \tag{26}$$

Denote $\mathcal{Z}_l{}^{new}$ by $\mathcal{Z}_m \tilde{\mathcal{Q}}_l$ and consider the reduced QR-factorization of $\hat{\mathcal{T}}_m \tilde{\mathcal{Q}}_l$ as

$$\hat{\mathcal{T}}_m \tilde{\mathcal{Q}}_l = \mathcal{Y}_l \hat{\mathcal{T}}_l^{new},$$

where $\mathcal{Y}_l \in \mathbb{C}^{(s_{m-1}+c_1)\times l}$ and $\hat{\mathcal{T}}_l^{new} \in \mathbb{C}^{l\times l}$. Thus, we have

$$A\mathcal{Z}_l^{new} = [\mathcal{W}_{m-1} \quad P_{m-1} \quad \hat{W}_m]\mathcal{Y}_l \hat{\mathcal{T}}_l^{new}. \tag{27}$$

Define

$$[\mathcal{W}_{l-1}^{new} \quad P_{l-1}^{new} \quad \hat{W}_l^{new}] = [\mathcal{W}_{m-1} \quad P_{m-1} \quad \hat{W}_m]\mathcal{Y}_l,$$

in which $\mathcal{W}_{l-1}^{new} \in \mathbb{C}^{n\times k}$ and $[P_{l-1}^{new} \quad \hat{W}_l^{new}] \in \mathbb{C}^{n\times c_1}$ are considered as the first k and the last c_1 columns of $[\mathcal{W}_{m-1} \quad P_{m-1} \quad \hat{W}_m]\mathcal{Y}_l$, respectively. Then, the relation (27) can be written as follows:

$$A\mathcal{Z}_l^{new} = [\mathcal{W}_{l-1}^{new} \quad P_{l-1}^{new} \quad \hat{W}_l]\hat{\mathcal{T}}_l^{new}. \tag{28}$$

For simplicity, we set $\mathcal{Z}_l = \mathcal{Z}_l^{new}$, $\mathcal{W}_{l-1} = \mathcal{W}_l^{new}$ and $P_{l-1} = P_{l-1}^{new}$, $\hat{W}_l = \hat{W}_l^{new}$ and $\mathcal{T}_l = \mathcal{T}_l^{new}$. Then, it can be shown that

$$A\mathcal{Z}_l = [\mathcal{W}_{l-1} \quad P_{l-1} \quad \hat{W}_l]\hat{\mathcal{T}}_l. \tag{29}$$

Consequently, from the relation (29), it turns out that the block Arnoldi relation holds.

Next, to obtain the deflated block Arnoldi relation, we define $\hat{\Lambda}_{l-1}$ as $\hat{\Lambda}_{l-1} = \hat{\mathcal{T}}_l(k+1 : k+c_1, :)$ and apply Algorithm 3 to determine the unitary matrix $G_l \in \mathbb{C}^{c_1\times c_1}$ and the numbers k_l and d_l such that $k_l + d_l = c_1$. Assume that the subspace decomposition is performed as follows

$$[W_l \quad P_l] = [P_{l-1} \quad \hat{W}_l]G_l. \tag{30}$$

Thus, using the relation (30), we can rewrite (29) as the following deflated block Arnoldi relation

$$\begin{aligned} A\mathcal{Z}_l &= [\mathcal{W}_{l-1} \quad P_{l-1} \quad \hat{W}_l]\mathcal{G}_l\mathcal{G}_l^H\hat{\mathcal{T}}_l \\ &= [\mathcal{W}_l \quad P_l]\mathcal{T}_l, \end{aligned} \tag{31}$$

where $\mathcal{G}_l = \begin{pmatrix} I_{s_{l-1}} & 0_{s_{l-1}\times c_1} \\ 0_{c_1\times s_{l-1}} & G_l \end{pmatrix}$. Now, in order to construct a new m-step deflated block Arnoldi relation, we assume that k is divisible by c_1, and accordingly, we allow that Steps 14 through 22 of Algorithm 4 are performed for $j = \frac{k}{c} + 1, \ldots, m$. As a consequence, the following relations obtain

$$\begin{aligned} &A\mathcal{Z}_m = [\mathcal{W}_m \quad P_m]\mathcal{T}_m, \\ &\mathcal{W}_m^H\mathcal{W}_m = I_{s_m}, \end{aligned}$$

where $\mathcal{Z}_m = [V_1 \quad \mathcal{W}_{m-1}]$. Then, after a cycle of SBGMRES-DR, the algorithm is restarted with the new initial guess $X_0^{new} = X_m$ and $R_0^{new} = R_m$ in which X_m and R_m are the approximate solution and residual derived from the previous cycle, respectively.

We end this subsection by presenting the next theorem which makes a practical way to update the new approximate solution and its corresponding residual.

Theorem 2 *Assume that $\mathcal{Z}_m = [V_1 \quad \mathcal{W}_{m-1}]$, $[\mathcal{W}_m \quad P_m]$ and $\mathcal{T}_m$ are obtained at the end of a cycle of the D-SBGMRES-DR method. In addition, let the deflated block Arnoldi relation be hold, i.e.,*

$$A\mathcal{Z}_m = [\mathcal{W}_m \quad P_m]\mathcal{T}_m.$$

Then, the approximate solution X_m^{new} is as follows

$$X_m^{new} = X_0^{new} + \mathcal{Z}_m^{new} Y_m^{new},$$

where $Y_m^{new} \in \mathbb{C}^{(s_{m-1}+c_1)\times p}$ satisfies the following matrix equation

$$\mathcal{T}_m^{new} Y_m^{new} = [0, \ldots, 0, (S_{l+1}^{new})^H, \ldots, (S_m^{new})^H, (P_m^{new})^H R_{m-1}^{new}]^H.$$

Proof After performing a cycle of D-SBGMRES-DR, the approximate residual R_m satisfies

$$R_m = R_0 - [\mathcal{W}_m \quad P_m][\mathcal{W}_m \quad P_m]^H R_0. \tag{32}$$

In the new cycle, we set $X_0^{new} = X_m$ and $R_0^{new} = R_m$. Thus, according to the relation (28), the deflated block Arnoldi relation can be expressed as $A\mathcal{Z}_m^{new} = [\mathcal{W}_m^{new} \quad P_l^{new}]\mathcal{T}_m^{new}$. Moreover, at the end of the cycle, the approximate solution X_m^{new} is of the form $X_m^{new} = X_0^{new} + \mathcal{Z}_m^{new} Y_m^{new}$, where $Y_m^{new} \in \mathbb{C}^{(s_{m-1}+c_1)\times p}$, and the corresponding residual R_m^{new} is

$$R_m^{new} = R_0^{new} - A\mathcal{Z}_m^{new} Y_m^{new} = R_0^{new} - [\mathcal{W}_m^{new} \quad P_m^{new}]\mathcal{T}_m^{new} Y_m^{new}.$$

Therefore, by applying the orthogonality condition $R_m^{new} \perp range\left([\mathcal{W}_m^{new} \quad P_m^{new}]\right)$, it follows that

$$\mathcal{T}_m^{new} Y_m^{new} = [\mathcal{W}_m^{new} \quad P_m^{new}]^H R_0^{new}. \tag{33}$$

As a direct consequence, we can verify that

$$\begin{aligned} R_m^{new} &= R_0^{new} - A\mathcal{Z}_m^{new} Y_m^{new} = R_0^{new} - [\mathcal{W}_m^{new} \quad P_m^{new}][\mathcal{W}_m^{new} \quad P_m^{new}]^H R_0^{new} \\ &= R_0^{new} - \mathcal{W}_{m-1}^{new}(\mathcal{W}_{m-1}^{new})^H R_0^{new} - W_m^{new}(W_m^{new})^H R_0^{new} - P_m^{new}(P_m^{new})^H R_0^{new} \\ &= R_{m-1}^{new} - W_m^{new} S_m^{new}, \end{aligned}$$

where $S_m^{new} = (W_m^{new})^H R_0^{new} = (W_m^{new})^H R_{m-1}^{new} \in \mathbb{C}^{k_m \times p}$.

On the other hand, since $R_0^{new} = R_m$, it is seen from the relation (32) that

$$[\mathcal{W}_l^{new} \quad P_l^{new}]^H R_0^{new} = [\mathcal{W}_l^{new} \quad P_l^{new}]^H (R_0 - [\mathcal{W}_m \quad P_m][\mathcal{W}_m \quad P_m]^H R_0). \tag{34}$$

Now, by substituting the following relations

$$\begin{aligned} [\mathcal{W}_l^{new} \quad P_l^{new}] &= [\mathcal{W}_{m-1}^{new} \quad P_{m-1}^{new} \quad \hat{W}_m^{new}]\mathcal{Y}_l \mathcal{G}_l, \\ [\mathcal{W}_m \quad P_m] &= [\mathcal{W}_{m-1}^{new} \quad P_{m-1}^{new} \quad \hat{W}_m^{new}]\mathcal{G}_m, \end{aligned}$$

into (34), we get

$$[\mathcal{W}_l^{new} \quad P_l^{new}]^H R_0^{new} = \mathcal{G}_l^H \mathcal{Y}_l^H [\mathcal{W}_{m-1}^{new} \quad P_{m-1}^{new} \quad \hat{W}_m^{new}]^H (R_0 - [\mathcal{W}_m \quad P_m][\mathcal{W}_m \quad P_m]^H R_0) = 0.$$

Consequently, we can conclude that $S_i^{new} = 0$, for $i = 1, 2, \ldots, l$, which is equivalent to say that $(\mathcal{W}_l^{new})^H R_0^{new} = 0$. Thus, the relation (33) can be written as follows

$$\mathcal{T}_m Y_m = [0, \ldots, 0, (S_{l+1}^{new})^H, (S_m^{new})^H, (P_m^{new})^H R_{m-1}^{new}]^H,$$

which completes the proof. □

4.2 Algorithm of D-SBGMRES-DR

The deflated simpler block GMRES method with vector deflation restarting (D-SBGMRES-DR) for solving (1) can be summarized in Algorithm 5.

Algorithm 5 The deflated SBGMRES method with vector deflation restarting (D-SBGMRES-DR).

Input: $A \in \mathbb{C}^{n\times n}$, $B \in \mathbb{C}^{n\times p}$. Choose a threshold $tol > 0$ and an initial guess $X_0 \in \mathbb{C}^{n\times p}$.
Output: $X_m \in \mathbb{C}^{n\times p}$.

1: Compute the initial guess block residual: $R_0 = B - AX_0$.
2: Use one cycle of the simpler block GMRES with vector deflation algorithm to determine the matrices $\mathcal{Z}_m = [V_1 \quad \mathcal{W}_{m-1}] \in \mathbb{C}^{n\times(s_{m-1}+c_1)}$ and $\hat{\mathcal{T}}_m \in \mathbb{C}^{(s_{m-1}+c_1)\times(s_{m-1}+c_1)}$, $[\mathcal{W}_{m-1} \quad P_{m-1} \quad \hat{W}_m] \in \mathbb{C}^{n\times(s_{m-1}+c_1)}$ such that

$$A\mathcal{Z}_m = [\mathcal{W}_{m-1} \quad P_{m-1} \quad \hat{W}_m]\hat{\mathcal{T}}_m.$$

Also, compute X_m and R_m.
3: **if** $\|R_m\|_F < tol$, **then** stop, **else** continue.
4: Define $X_0 = X_m$ and $R_0 = R_m$.
5: Compute the eigenpairs of the generalized eigenvalue problem (23) or (25) by using the QZ algorithm. Let $\tilde{g}_1, \tilde{g}_2, \ldots, \tilde{g}_l$ be the eigenvectors associated with the $l = k + c_1$ smallest eigenvalues magnitude of (23) or (25). Define $\tilde{\mathcal{G}}_l = [\tilde{g}_1, \tilde{g}_2, \ldots, \tilde{g}_l]$, we first separate the $\tilde{g}_i$'s into real and imaginary parts if they are complex, to form the columns of $\tilde{\mathcal{G}}_l \in \mathbb{R}^{(s_{m-1}+c_1)\times l}$. Both the real and the imaginary parts need to be included. Then, we compute the reduced QR-factorization of $\tilde{\mathcal{G}}_l \in \mathbb{R}^{(s_{m-1}+c_1)\times l}$ as $\tilde{\mathcal{G}}_l = \tilde{\mathcal{Q}}_l\tilde{\mathcal{R}}_l$, where $\tilde{\mathcal{Q}}_l \in \mathbb{R}^{(s_{m-1}+c_1)\times l}$ and $\tilde{\mathcal{R}}_l \in \mathbb{R}^{l\times l}$.
6: Set $\mathcal{Z}_l^{new} = \mathcal{Z}_m\tilde{\mathcal{Q}}_l$.
7: Compute the reduced QR-factorization of $\hat{\mathcal{T}}_m\tilde{\mathcal{Q}}_l$ as $\hat{\mathcal{T}}_m\tilde{\mathcal{Q}}_l = \mathcal{Y}_l\mathcal{T}_l^{new}$, where $\mathcal{Y}_l \in \mathbb{C}^{(s_{m-1}+c_1)\times l}$ and $\hat{\mathcal{T}}_l^{new} \in \mathbb{C}^{l\times l}$.
8: Define $[\mathcal{W}_{l-1}^{new} \quad P_{l-1}^{new} \quad \hat{W}_l^{new}] = [\mathcal{W}_{m-1} \quad P_{m-1} \quad \hat{W}_m]\mathcal{Y}_l$ with $\mathcal{W}_{l-1}^{new} \in \mathbb{R}^{n\times k}$ and $[P_{l-1}^{new} \quad \hat{W}_l^{new}] \in \mathbb{C}^{n\times c_1}$ as the first k and the last c_1 columns of $[\mathcal{W}_{m-1} \quad P_{m-1} \quad \hat{W}_m]\mathcal{Y}_l$, respectively. Thus the deflated block Arnoldi relation holds:

$$A\mathcal{Z}_l^{new} = [\mathcal{W}_{l-1}^{new} \quad P_{l-1}^{new} \quad \hat{W}_l^{new}]\hat{\mathcal{T}}_l^{new}.$$

9: Define $\mathcal{Z}_l = \mathcal{Z}_l^{new}$, $[\mathcal{W}_{l-1} \quad P_{l-1} \quad \hat{W}_l] = [\mathcal{W}_l^{new} \quad P_l^{new} \quad \hat{W}_l^{new}]$, $\hat{\mathcal{T}}_l = \hat{\mathcal{T}}_l^{new}$ and $\hat{\Lambda}_{l-1} = \hat{\mathcal{T}}_l(k+1 : k+c_1, :)$.
10: Apply Algorithm 3 to determine the unitary matrix $G_l \in \mathbb{C}^{c_1\times c_1}$ and the numbers k_l, d_l such that $k_l + d_l = c_1$.
11: Define $[\mathcal{W}_l \quad P_l] = [\mathcal{W}_{l-1} \quad P_{l-1} \quad \hat{W}_l]\mathcal{G}_l$ and $\mathcal{T}_l = \mathcal{G}_l^H\hat{\mathcal{T}}_l$, where $\mathcal{G}_l = \begin{pmatrix} I_{s_{l-1}} & 0_{s_{l-1}\times c_1} \\ 0_{c_1\times s_{l-1}} & G_l \end{pmatrix}$.
12: Set $R_l = R_0$ and $s(\frac{k}{c_1} + 1) = k$.
13: Run Algorithm 2 for $j = \frac{k}{c_1} + 1, \ldots, m$. Then, form the new approximate solution $X_j = X_0 + \mathcal{Z}_jY_j$ and compute the residual matrix $R_j = B - AX_j$.
14: Check convergence. If the convergence criterion is not met, then set $X_0 = X_j$ and $R_0 = R_j$, and go to 5.

5 A Theoretical Result for SBGMRES-DR and D-SBGMRES-DR

Calandra et al. [7] studied the effects of the number of selected Krylov directions on the computational costs of BGMRES with vector deflation restarting. To be more specific, this important issue was analyzed that the norm minimization property of BGMRES with vector deflation restarting creates the conditions under which the number of Krylov directions will decrease, and accordingly, the number of matrix–vector products can be reduced noticeably. In a similar way, Meng et al. [29] also investigated this result for their proposed deflated method.

Following up these researches, we intend to figure out the information about the number of Krylov directions employed in the SBGMRES-DR and D-SBGMRES-DR methods. As explained in previous sections, the number of Krylov directions k_j selected for SBGMRES-DR and D-SBGMRES-DR is determined based on the amounts of singular values of $\hat{\Lambda}_{j-1}$. For this reason, a theorem is provided in the rest of this section to present the behavior of the singular values of $\hat{\Lambda}_{j-1}$ obtained from two consecutive iterations of a cycle of the SBGMRES-DR and D-SBGMRES-DR methods.

Theorem 3 *Suppose that the matrices*

$$\hat{\mathcal{T}}_j = \begin{bmatrix} \mathcal{T}_{j-1} & T_j \\ 0_{k_{j-1}\times(s_{j-2}+c_1)} & T_{jj} \end{bmatrix}, \qquad \hat{\Lambda}_{j-1} = \begin{bmatrix} (\mathcal{T}_{j-1})^- & (T_j)^- \\ 0_{k_{j-1}\times(s_{j-2}+c_1)} & T_{jj} \end{bmatrix},$$
$$\hat{\mathcal{T}}_{j+1} = \begin{bmatrix} \mathcal{T}_j & T_{j+1} \\ 0_{k_j\times(s_{j-1}+c_1)} & T_{j+1,j+1} \end{bmatrix}, \qquad \hat{\Lambda}_j = \begin{bmatrix} (\mathcal{T}_j)^- & (T_{j+1})^- \\ 0_{k_j\times(s_{j-1}+c_1)} & T_{j+1,j+1} \end{bmatrix},$$

are generated at the j-th and the $(j+1)$-th iterations of a given cycle of the SBGMRES-DR and D-SBGMRES-DR methods, respectively. In addition, assume that the singular values of $\left[(\mathcal{T}_{j-1})^+ \;\; (T_j)^+\right]$ are arranged in decreasing order:

$$\sigma_{max} = \sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_{s_{j-1}} = \sigma_{min},$$

and $\sigma_{1-k_j}\left(\left[(\mathcal{T}_{j-1})^+ \;\; (T_j)^+\right]\right) = 0$. Then, the following relations for both the SBGMRES-DR and D-SBGMRES-DR methods are satisfied:

$$\sigma_i\left(\hat{\Lambda}_j\right) \leq \sigma_i\left(\hat{\Lambda}_{j-1}\right), \quad \text{for } i = 1, 2, \ldots, c_1. \tag{35}$$

Proof The proof for SBGMRES-DR starts with recalling the relation (19) that $\mathcal{T}_j = (\mathcal{G}_j)^H \hat{\mathcal{T}}_j$, where $\mathcal{G}_j \in \mathbb{C}^{(s_{j-1}+c_1)\times(s_{j-1}+c_1)}$ is a unitary matrix. By utilizing [44, Theorem 8.13], it can be seen that

$$\sigma_i\left(\mathcal{T}_j\right) \leq \sigma_i\left((\mathcal{G}_j)^H\right)\sigma_i\left(\hat{\mathcal{T}}_j\right),$$

for $i = 1, 2, \ldots, s_{j-1} + c_1$. Since $(\mathcal{G}_j)^H \mathcal{G}_j = I$, it follows that

$$\sigma_i\left(\mathcal{T}_j\right) \leq \sigma_i\left(\hat{\mathcal{T}}_j\right), \quad \text{for } i = 1, 2, \ldots, s_{j-1} + c_1. \tag{36}$$

On the other hand, let us assume that the matrices $\hat{\mathcal{T}}_j$ and $\hat{\mathcal{T}}_{j+1}$ are partitioned as

$$\hat{\mathcal{T}}_j = \begin{bmatrix} (\mathcal{T}_{j-1})^+ & (T_j)^+ \\ \multicolumn{2}{c}{\hat{\Lambda}_{j-1}} \end{bmatrix}, \quad \hat{\mathcal{T}}_{j+1} = \begin{bmatrix} (\mathcal{T}_j)^+ & (T_{j+1})^+ \\ \multicolumn{2}{c}{\hat{\Lambda}_j} \end{bmatrix},$$

in which $(\mathcal{T}_{j-1})^+ = \mathcal{T}_{j-1}(1 : s_{j-1}, 1 : s_{j-2} + c_1)$, $(T_j)^+ = T_j(1 : s_{j-1}, 1 : k_{j-1})$, $(\mathcal{T}_j)^+ = \mathcal{T}_j(1 : s_j, 1 : s_{j-1} + c_1)$, and $(T_{j+1})^+ = T_{j+1}(1 : s_j, 1 : k_j)$. Now, according to

[44, Theorem 8.12], we derive

$$\lambda_i\Big((\hat{\mathcal{T}}_{j+1})^H\hat{\mathcal{T}}_{j+1}\Big) \geq \lambda_i\Big((\hat{\Lambda}_j)^H\hat{\Lambda}_j\Big),$$

and

$$\lambda_i\Big((\hat{\mathcal{T}}_j)^H\hat{\mathcal{T}}_j\Big) \leq \lambda_i\Big((\hat{\Lambda}_{j-1})^H\hat{\Lambda}_{j-1}\Big) + \lambda_1\Big(\big[(\mathcal{T}_{j-1})^+ \ \ (T_j)^+\big]^H\big[(\mathcal{T}_{j-1})^+ \ \ (T_j)^+\big]\Big),$$

for $i = 1, 2, \ldots, c_1$. Consequently, it turns out that

$$\sigma_i\Big(\hat{\mathcal{T}}_{j+1}\Big) \geq \sigma_i\Big(\hat{\Lambda}_j\Big), \quad i = 1, 2, \ldots, c_1, \tag{37}$$

and

$$\sigma_i\Big(\hat{\mathcal{T}}_j\Big) \leq \sigma_i\Big(\hat{\Lambda}_{j-1}\Big) + \sigma_1\Big(\big[(\mathcal{T}_{j-1})^+ \ \ (T_j)^+\big]\Big), \quad i = 1, 2, \ldots, c_1. \tag{38}$$

Furthermore, in view of [44, Theorem 8.14], we get

$$\sigma_{k_j+i}\Big(\hat{\mathcal{T}}_{j+1}\Big) \leq \sigma_i\Big(\mathcal{T}_j\Big), \quad i = 1, 2, \ldots, s_{j-1} + c_1. \tag{39}$$

By considering the relations (36), (37), (38) and (39), the following inequalities hold

$$\sigma_i\Big(\hat{\Lambda}_j\Big) \leq \sigma_i\Big(\hat{\Lambda}_{j-1}\Big) + \sigma_{1-k_j}\Big(\big[(\mathcal{T}_{j-1})^+ \ \ (T_j)^+\big]\Big), \quad i = 1, 2, \ldots, c_1,$$

and the proof for the case of SBGMRES-DR is finished.

The rest of the proof for D-SBGMRES-DR is analogous with that for SBGMRES-DR and is omitted. □

To sum up, the above-mentioned theorem demonstrates that the singular values of $\hat{\Lambda}_{j-1}$, constructed from two consecutive iterations of a cycle of SBGMRES-DR and D-SBGMRES-DR, are on the decline in a monotone manner. In Algorithm 3, it was identified that there is a close relationship between the number of chosen Krylov directions (i.e., k_j) and the amount of singular values of $\hat{\Lambda}_{j-1}$. In point of fact, it was stated that $k_j + d_j = c_1$. For this reason, it is expected that the declining trend in the amount of singular values of $\hat{\Lambda}_{j-1}$ leads to a reduction in the values of k_j, and consequently the cost of matrix–vector products will be also reduced.

6 Computational Cost of D-SBGMRES-DR

In this section, the total computational cost of a given cycle of the D-SBGMRES-DR method is evaluated. To this end, let us suppose that the QR-factorization and SVD are based on the modified Gram-Schmidt algorithm and the Golub-Reinsch algorithm, respectively; for further details about the Golub-Reinsch algorithm, see [19]. For an $n \times m$ matrix R, the QR-factorization and the SVD factorization require $2mn^2$ and $4nm^2 + 8m^3$ operations, respectively. The deflated block Arnoldi process also costs C_j operations, where

$$C_j = 2n^2k_{j-1} + njk_{j-1} + 4nd_{j-1}k_{j-1} + 2nk_{j-1}^2 + \sum_{i=1}^{j-1} 4nk_ik_{j-1}.$$

Moreover, it is necessary to take into account the computational cost of $[W_j \ P_j]$, G_j, $\hat{\Lambda}_{j-1}$ and $\mathcal{T}_j$ along with other computational considerations. To summarize, the details about the computational costs for a cycle of the D-SBGMRES-DR method are reported in Table 1. It is

Table 1 Computational costs of a given cycle of the D-SBGMRES-DR method

Operation	Computational cost
QR-factorization	$2np^2 + 4p^3 + 2nc_1^2$
Deflated block Arnoldi process	C_j
Computation of V_1	$4p^3 + 12p^3 + 2npc_1$
Computation of G_1	$16c_1^3$
Computation of $[W_1 \quad P_1]$	$2nc_1^2$
Computation of $\mathcal{T}_1$	$2c_1^3$
Computation of S_1	$2npk_1$
Computation of R_1	$2npk_1 + np$
Computation of G_j in Line 2 of Algorithm 5	$\Sigma_{j=2}^{m} 8(s_{j-1} + c_1)^3 + 8c_1(s_{j-1} + c_1)^2$
Computation of G_j in Line 12 of Algorithm 5	$\Sigma_{j=\frac{k}{c_1}+1}^{m} 8(s_{j-1} + c_1)^3 + 8c_1(s_{j-1} + c_1)^2$
Computation of $[W_j \quad P_j]$	$2nc_1^2(m-1)$ or $2nc_1^2(m - \frac{k}{c_1})$
Computation of $\mathcal{T}_j$	$\Sigma_{j=2}^{m} 2c_1^2(s_{j-1} + c_1)$ or $\sum_{j=\frac{k}{c_1}+1}^{m} 2c_1^2(s_{j-1} + c_1)$
Computation of S_j	$\sum_{j=2}^{m} 2npk_j$ or $\sum_{j=\frac{k}{c_1}+1}^{m} 2npk_j$
Computation of R_j	$\sum_{j=2}^{m} 2npk_j$ or $\sum_{j=\frac{k}{c_1}+1}^{m} 2npk_j$

observed from this table that the computational cost of $[W_j \ P_j]$ is quite expensive, whereas the computational costs of some major items, such as G_j and $\hat{\Lambda}_{j-1}$ (and therefor, $G_j^H \hat{\Lambda}_{j-1}$), are relatively inexpensive, due to the fact that the size of these matrices does not depend on n.

7 Numerical Examples

In this section, some numerical experiments are conducted to evaluate the performance of the proposed methods SBGMRES-DR and D-SBGMRES-DR. In addition, the numerical behavior of SBGMRES-DR and D-SBGMRES-DR is compared with that of some block Krylov subspace methods whose description is given in Table 2.

In what follows, the zero matrix $X_0 = 0_{n\times p}$ is considered as the initial guess, k denotes the number of harmonic Ritz vectors, m is the maximum dimension of the Krylov subspace, and ϵ_d indicates the deflation threshold. All the methods are evaluated in terms of the three key criteria: the number of matrix–vector products (referred to Mvps), the total running time in seconds (referred to CPU) and the real relative residual norm with respect to the Frobenius norm (referred to res.norm):

$$\text{res.norm} = \frac{\|B - AX_m\|_F}{\|B\|_F}. \tag{40}$$

Moreover, a test is stopped as soon as the maximum number of restart cycles is more than 2000, or the condition $\|B - AX_m\|_F < 10^{-6}\|B\|_F$ is satisfied. All the numerical results were carried out in Matlab Release 2015b, on a PC with an Intel(R) Core(TM) i5 CPU, 4G bytes of memory and 2.00 GHz frequency.

Table 2 Description of the block Krylov subspace methods used in the experiments

Method	Abbreviation
The deflated simpler block GMRES method with vector deflation restarting	D-SBGMRES-DR (Ours)
The simpler block GMRES method with vector deflation restarting	SBGMRES-DR (Ours)
The modified block GMRES method with deflation at each iteration [7]	BGMRES-S
The residual-based simpler block GMRES method with deflation [42]	RB-SBGMRES-D
The restarted block GMRES method with deflation of eigenvalues [30]	BGMRES-DR
The adaptive simpler block GMRES method with deflated restarting [46]	Ad-SBGMRES-DR
The loop-interchange block GMRES [17]	Li-BGMRES
The simpler block CMRH [1]	SBCMRH
The simpler block GMRES [27]	SBGMRES
The block GMRES [8]	BGMRES

Table 3 Generic properties of some test matrices used in Examples 1 to 5

Matrix	n	nnz	Application area
fv3	9801	87,025	2D/3D problem
Dubcova1	16,129	253,009	2D/3D problem
wathen100	30,401	471,601	Random 2D/3D problem
wathen120	36,441	565,761	Random 2D/3D problem
Dubcova2	65,025	1,030,225	2D/3D problem
Dubcova3	146,689	3,636,643	2D/3D problem
FEM_3D_thermal2	147,900	3,489,300	Thermal problem
cage13	445,315	7,479,343	Directed weighted graph
cage14	1,505,785	27,130,349	Directed weighted graph
tridiag	1000	2998	Academic model
bidiag	1000	1999	Academic model

In Examples 1 to 5, two sets of test matrices are considered. The first one contains nine matrices which are selected from the University of Florida Sparse Matrix Collection [12], and the second one contains two matrices "tridiag" and "bidiag" that are taken from [5,30]. The matrix tridiag is tridiagonal in which the diagonal elements are $0.1, 0.2, 0.3, 0.4, 0.5, 6, 7, \ldots, 1000$, and the sub and the super diagonal elements being all 1. The matrix bidiag is bidiagonal with super diagonal elements being all 0.1, and its main diagonal elements are $1, 2, 3, \ldots, 1000$. In Example 6, the efficiency of the proposed methods SBGMRES-DR and D-SBGMRES-DR and the other compared methods to solve the two-dimensional Helmholtz equations is examined. Generic properties of the test matrices used in the experiments are also given in Table 3.

Example 1 In this example, we make a comparison between the performance of the SBGMRES-DR and D-SBGMRES-DR methods and that of RB-SBGMRES-D, BGMRES-S, Ad-SBGMRES-DR and BGMRES-DR. Here, it should be noted that the SBGMRES-DR, RB-SBGMRES-D and BGMRES-S methods are based on the vector deflation technique, while the BGMRES-DR and Ad-SBGMRES-DR methods are based on the eigenvalue deflation technique. In addition, the D-SBGMRES-DR method benefits from both the vector and eigenvalue deflation techniques. This set of experiments is performed with $m = 30$, $k = 10$, $p = 10$ and $\epsilon_d = 10^{-3}$. Moreover, the right-hand side matrix B is considered as $B = \mathtt{rand}(n, p)$.

Based on the performance results reported in Table 4, it is observed that D-SBGMRES-DR needs a smaller amount of matrix–vector products and CPU time in comparison with the other methods. In particular, D-SBGMRES-DR is greatly superior to the vector deflation methods SBGMRES-DR, BGMRES-S and RB-SBGMRES-D. Moreover, all the methods reach almost the same accuracy in terms of the relative residual norm. Therefore, the obtained results declare that D-SBGMRES-DR has a high potential to produce favorable outcomes, and it is also more robust and efficient than the other methods.

The convergence curves of the current methods are displayed in Fig. 1 which presents the relative residual norms with respect to the number of matrix–vector products.

It should be mentioned that in Fig. 1, the curves of the BGMRES-DR and Ad-SBGMRES-DR methods are overlapped with each other in all cases. It is appeared from this figure that the D-SBGMRES-DR, BGMRES-DR and Ad-SBGMRES-DR methods require fewer matrix–vector products than the other methods in order to converge to the expected accuracy, and D-SBGMRES-DR works better than BGMRES-DR and Ad-SBGMRES-DR. Moreover, the D-SBGMRES-DR method attains a smaller relative residual norm than the other methods in many cases. Therefore, we may infer that D-SBGMRES-DR is effective in practice, and the combination of eigenvalue and vector deflation techniques is a major benefit of the D-SBGMRES-DR method.

Example 2 In this example, in order to assess the efficiency of the D-SBGMRES-DR method, a comparison between this method and SBGMRES-DR is drawn. The numerical results for $p = 4, 8, 12$, $m = 20, 30$, $k = 24$, and $\epsilon_d = 10^{-3}$ are collected in Table 5. We also assume that $B = \mathtt{rand}(n, p)$.

It is seen from Table 5 that D-SBGMRES-DR outperforms SBGMRES-DR in terms of the number of matrix–vector products and CPU time in all cases. In addition, except for SBGMRES-DR on the test matrix tridiag, when the number of right-hand sides p increases from 4 to 8 and 12, the number of matrix–vector products of the D-SBGMRES-DR and SBGMRES-DR methods is on the rise for all cases. It is also deduced that for the test matrix Dubcova3 with $m = 20$ and $p = 4$, both D-SBGMRES-DR and SBGMRES-DR do not converge within 2000 restart cycles, while the convergence of these methods is much improved for the case of $m = 30$ and the other values of p.

Example 3 The aim of this example is to examine the impact of the deflation threshold parameter on the convergence of the D-SBGMRES-DR, SBGMRES-DR, RB-SBGMRES-D and BGMRES-S methods. Here, the deflation threshold parameter is chosen as $\epsilon_d = 1, 0.1$, 0.01, and we consider $m = 20$, $k = 10$ and $p = 5$. Moreover, we let $B = \mathtt{rand}(n, p)$. Table 6 reports the performance results of the methods on the different values of ϵ_d.

It is perceived from Table 6 that except for the cases of "the D-SBGMRES-DR method on the test matrix wathen120" and "the BGMRES-S method on the test matrix tridiag", the number of matrix–vector products and CPU time for the other cases increase when the value of ϵ_d decreases from 1 to 0.01. However, the methods D-SBGMRES-DR (for the test matrix wathen120 when $\epsilon_d = 0.01$) and BGMRES-S (for the test matrix tridiag when $\epsilon_d = 0.1$) converge in less matrix–vector products and CPU time. On the other hand, the number of matrix–vector products required for the convergence of D-SBGMRES-DR does not change dramatically with respect to the variations of the deflation threshold parameter ϵ_d for the test matrices fv3, Dubcova1, Dubcova3 and bidiag. Moreover, the D-SBGMRES-DR method is

Table 4 Example 1: Performance results for the methods D-SBGMRES-DR, SBGMRES-DR, BGMRES-S, RB-SBGMRES-D, BGMRES-DR, and Ad-SBGMRES-DR

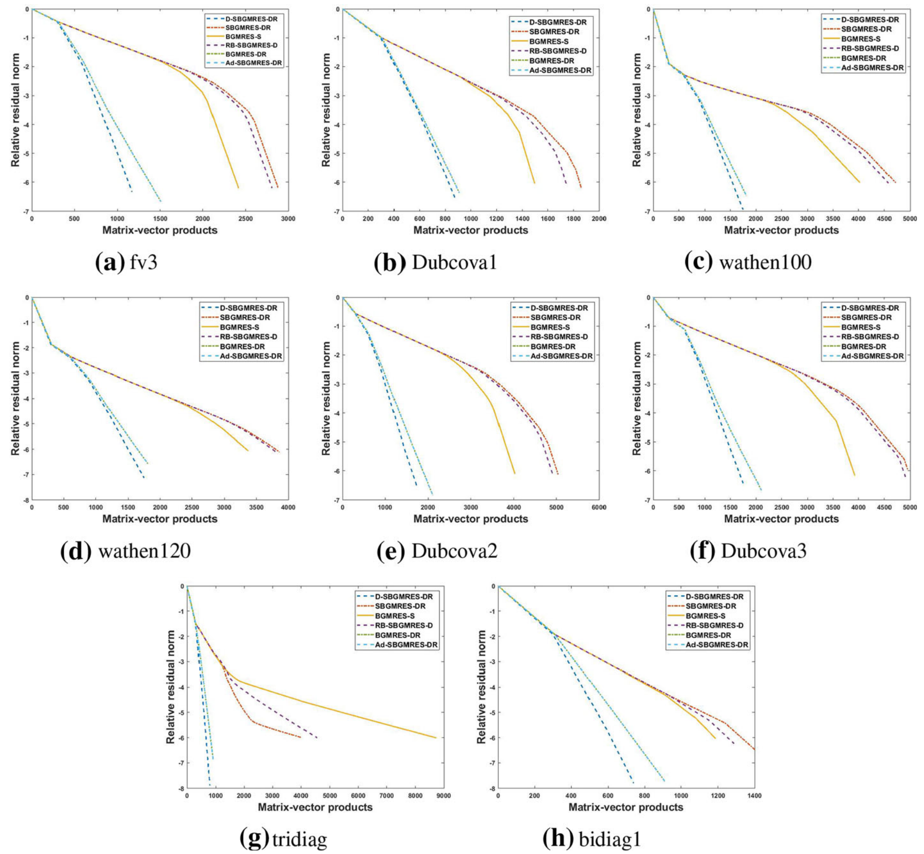
Method	fv3			Dubcova1		
	Mvps	CPU	res.norm	Mvps	CPU	res.norm
D-SBGMRES-DR(m, k)	**1170**	**5.9440**	4.7640e−07	**880**	**6.2652**	**2.7127e−07**
SBGMRES-DR(m)	2880	7.1328	6.3230e−07	1860	9.0784	6.4345e−07
BGMRES-S(m)	2414	8.8895	6.5327e−07	1496	9.3442	9.2070e−07
RB-SBGMRES-D(m)	2807	8.5182	6.4581e−07	1749	10.385	7.1147e−07
BGMRES-DR(m, k)	1800	6.9034	**2.1833e−07**	1200	9.7008	4.3956e−07
Ad-SBGMRES-DR(m, k)	1510	6.0640	**2.1833e−07**	910	8.6202	4.3956e−07
Method	wathen100			wathen120		
	Mvps	CPU	res.norm	Mvps	CPU	res.norm
D-SBGMRES-DR(m, k)	**1750**	**24.174**	**1.1451e−07**	**1750**	**29.489**	**7.2573e−08**
SBGMRES-DR(m)	4720	43.980	9.5277e−07	3860	48.573	7.7580e−07
BGMRES-S(m)	4020	40.364	9.8526e−07	3368	45.853	8.7872e−07
RB-SBGMRES-D(m)	4583	51.979	9.3908e−07	3809	53.919	7.7237e−07
BGMRES-DR(m, k)	2100	32.178	3.5290e−07	2100	38.668	2.6391e−07
Ad-SBGMRES-DR(m,k)	1810	30.427	3.5290e−07	1810	35.812	2.6391e−07
Method	Dubcova2			Dubcova3		
	Mvps	CPU	res.norm	Mvps	CPU	res.norm
D-SBGMRES-DR(m, k)	**1750**	**62.824**	2.5093e−07	**1750**	**146.48**	3.7016e−07
SBGMRES-DR(m)	5040	120.92	7.4680e−07	4960	278.32	9.9642e−07
BGMRES-S(m)	4023	94.938	8.0711e−07	3923	217.65	7.1355e−07
RB-SBGMRES-D(m)	4912	141.88	7.6240e−07	4907	336.52	6.2100e−07
BGMRES-DR(m, k)	2400	79.198	**1.4565e−07**	2400	191.08	**2.1532e−07**
Ad-SBGMRES-DR(m,k)	2110	75.262	**1.4565e−07**	2110	167.77	**2.1532e−07**
Method	tridiag			bidiag		
	Mvps	CPU	res.norm	Mvps	CPU	res.norm
D-SBGMRES-DR(m, k)	**790**	**1.3672**	**1.3040e−08**	**740**	**1.4302**	**1.6293e−08**
SBGMRES-DR(m)	4022	4.5331	9.9824e−07	1400	1.4762	3.3834e−07
BGMRES-S(m)	8716	17.630	9.8357e-07	1185	2.9400	9.3959e−07
RB-SBGMRES-D(m)	4552	5.7257	9.8239e−07	1287	1.6308	5.8224e-07
BGMRES-DR(m, k)	1200	1.8412	1.2664e−07	1200	1.6325	1.8689e−08
Ad-SBGMRES-DR(m, k)	910	1.7862	1.2664e−07	910	1.5796	1.8689e−08

Here, $m = 30$, $k = 10$, $p = 10$ and $\epsilon_d = 10^{-3}$. The best results are marked in bold

of higher quality than the other methods in terms of the number of matrix–vector products and CPU time.

In order to carry out a further investigation into the use of the vector deflation strategy, the variations of k_j in terms of the number of matrix–vector products are displayed in Fig. 2.

Figure 2 demonstrates that the SBGMRES-DR, RB-SBGMRES-DR and BGMRES-S methods have a decreasing behavior for the different values of k_j. On the other hand, the



(a) fv3 **(b)** Dubcova1 **(c)** wathen100

(d) wathen120 **(e)** Dubcova2 **(f)** Dubcova3

(g) tridiag **(h)** bidiag1

Fig. 1 Example 1: Convergence curves of D-SBGMRES-DR, SBGMRES-DR, BGMRES-S, RB-SBGMRES-D, BGMRES-DR, and Ad-SBGMRES-DR

plots associated to the D-SBGMRES-DR method first follow a decreasing trend for k_j, then they experience a rising one. Therefore, we can conclude that the computational cost of the D-SBGMRES-DR method can be reduced. A possible explanation for this issue is that D-SBGMRES-DR takes the advantage of utilizing spectral information at a restart cycle.

Example 4 In this example, the effectiveness of our proposed methods and the other eight block Krylov subspace methods presented in Table 2 is evaluated. In this regard, the four test matrices fv3, Dubcova1, tridiag and bidiag are chosen to perform the task of comparison. Moreover, the dimension of the block Krylov subspace m is selected as 20, 30, and the parameters k, p, and ϵ_d are set to be 10, 10, and 1, respectively. The right-hand side matrix B is also taken as $B = \mathtt{rand}(n, p)$. The numerical results for the different values of m are reported in Table 7.

Table 7 shows that the D-SBGMRES-DR method works much better than the other methods according to the number of matrix–vector products and CPU time. Moreover, it can be seen that in the most cases, D-SBGMRES-DR produces more satisfactory results with respect to the relative residual norm.

Table 5 Example 2: performance results for the D-SBGMRES-DR and SBGMRES-DR methods

$m = 20$		$p = 4$		$p = 8$		$p = 12$	
Test matrix	Method	Mvps	CPU	Mvps	CPU	Mvps	CPU
fv3	D-SBGMRES-DR(m, k)	**528**	**2.6382**	**976**	**5.0495**	**1320**	**7.4890**
	SBGMRES-DR(m)	2288	4.1486	3464	7.1945	4324	10.226
Dubcova1	D-SBGMRES-DR(m, k)	**416**	**3.2757**	**840**	**6.8580**	**1104**	**10.605**
	SBGMRES-DR(m)	1252	3.7599	1925	7.2052	2376	11.203
wathen100	D-SBGMRES-DR(m, k)	**808**	**12.325**	**1384**	**24.300**	**1968**	**36.125**
	SBGMRES-DR(m)	4012	22.828	5600	41.445	6944	57.405
wathen120	D-SBGMRES-DR(m, k)	**808**	**16.370**	**1384**	**28.785**	**1968**	**41.633**
	SBGMRES-DR(m)	2604	21.844	4500	45.589	5781	63.266
Dubcova2	D-SBGMRES-DR(m, k)	**1032**	**40.292**	**1384**	**50.756**	**1968**	**76.475**
	SBGMRES-DR(m)	3956	67.299	6164	116.62	7757	159.84
Dubcova3	D-SBGMRES-DR(m, k)	–	–	**1384**	**113.10**	**1968**	**167.14**
	SBGMRES-DR(m)	–	–	6116	275.08	7532	359.97
Tridiag	D-SBGMRES-DR(m, k)	**416**	**0.7181**	**568**	**0.9168**	**852**	**1.5822**
	SBGMRES-DR(m)	25096	40.021	2396	2.5426	5696	6.1135
Bidiag	D-SBGMRES-DR(m, k)	**360**	**0.6351**	**568**	**0.9048**	**672**	**1.2079**
	SBGMRES-DR(m)	968	1.2142	1392	1.3795	1756	1.6845
$m = 30$		$p = 4$		$p = 8$		$p = 12$	
Test matrix	Method	Mvps	CPU	Mvps	CPU	Mvps	CPU
fv3	D-SBGMRES-DR(m, k)	**600**	**3.5002**	**888**	**5.5341**	**1368**	**10.202**
	SBGMRES-DR(m)	1722	3.7792	2604	6.9452	3294	10.262
Dubcova1	D-SBGMRES-DR(m, k)	**408**	**3.4727**	**672**	**6.3797**	**1032**	**12.273**
	SBGMRES-DR(m)	1060	3.7451	1602	7.2260	2088	12.302
wathen100	D-SBGMRES-DR(m, k)	**792**	**14.614**	**1320**	**28.658**	**2040**	**48.841**
	SBGMRES-DR(m)	2560	18.277	4132	40.595	5214	56.327
wathen120	D-SBGMRES-DR(m, k)	**792**	**19.254**	**1320**	**33.710**	**1704**	**46.312**
	SBGMRES-DR(m)	1860	20.090	3322	45.240	4428	64.106
Dubcova2	D-SBGMRES-DR(m, k)	**792**	**35.749**	**1320**	**60.296**	**2040**	**98.673**
	SBGMRES-DR(m)	2910	64.175	4560	111.39	5490	143.38
Dubcova3	D-SBGMRES-DR(m, k)	**792**	**83.087**	**1536**	**233.96**	**2040**	**342.22**
	SBGMRES-DR(m)	2680	145.13	4340	377.23	5460	516.65
Tridiag	D-SBGMRES-DR(m, k)	**408**	**0.7886**	**616**	**1.2829**	**696**	**1.7185**
	SBGMRES-DR(m)	2904	3.7639	2220	2.7498	1692	2.1290
Bidiag	D-SBGMRES-DR(m, k)	**392**	**0.7689**	**584**	**1.2379**	**696**	**1.7046**
	SBGMRES-DR(m)	838	1.1231	1278	1.5231	1398	1.7609

Here, $p = 4, 8, 12$, $m = 20, 30$, $k = 24$, and $\epsilon_d = 10^{-3}$. The best results are marked in bold. In addition, the notation "–" shows that the method cannot converge after 2000 restart cycles

Table 6 Example 3: performance results for the D-SBGMRES-DR, SBGMRES-DR, RB-SBGMRES-D and BGMRES-S methods

Test matrix	Method	$\epsilon_d = 1$		$\epsilon_d = 0.1$		$\epsilon_d = 0.01$	
		Mvps	CPU	Mvps	CPU	Mvps	CPU
fv3	**D-SBGMRES-DR(m, k)**	**730**	**3.2717**	730	3.2913	730	3.2721
	SBGMRES-DR(m)	**2070**	**4.0761**	2290	4.5888	2425	4.6910
	BGMRES-S(m)	**1754**	**4.0775**	1907	4.4873	2085	4.8524
	RB-SBGMRES-D(m)	**2034**	**5.1249**	2243	5.3881	2389	5.5839
Dubcova1	**D-SBGMRES-DR(m, k)**	**550**	**3.9237**	550	3.9888	550	3.9237
	SBGMRES-DR(m)	**1150**	**3.9968**	1271	4.0816	1350	4.4544
	BGMRES-S(m)	**937**	**3.9962**	1037	3.9987	1115	4.0807
	RB-SBGMRES-D(m)	**1112**	**4.4489**	1236	4.9761	1310	5.1207
wathen100	**D-SBGMRES-DR(m, k)**	**1000**	**15.361**	1090	15.535	1000	15.438
	SBGMRES-DR(m)	**3765**	**24.780**	4050	27.219	4315	28.422
	BGMRES-S(m)	**3153**	**21.804**	3388	23.468	3781	25.149
	RB-SBGMRES-D(m)	**3726**	**30.555**	4015	33.283	4270	33.913
wathen120	**D-SBGMRES-DR(m, k)**	1090	20.427	1090	20.661	**1000**	**18.951**
	SBGMRES-DR(m)	**2995**	**28.108**	3120	29.836	3220	31.580
	BGMRES-S(m)	**2512**	**22.740**	2861	26.494	3165	30.761
	RB-SBGMRES-D(m)	**2977**	**31.517**	3115	33.514	3206	35.907
Dubcova2	**D-SBGMRES-DR(m, k)**	**1090**	**38.886**	1090	39.242	1180	42.835
	SBGMRES-DR(m)	**3685**	**67.055**	3905	70.769	4245	79.449
	BGMRES-S(m)	**2998**	**52.792**	3152	61.342	3495	65.320
	RB-SBGMRES-D(m)	**3629**	**79.936**	3872	87.117	4212	93.329
Dubcova3	**D-SBGMRES-DR(m, k)**	**1090**	**90.031**	1090	90.411	1090	90.498
	SBGMRES-DR(m)	**3495**	**157.40**	3811	172.83	4210	196.31
	BGMRES-S(m)	**2793**	**124.72**	3047	136.06	3436	156.87
	RB-SBGMRES-D(m)	**3467**	**194.87**	3816	212.88	4184	236.17
tridiag	**D-SBGMRES-DR(m, k)**	**460**	**0.7092**	550	0.8343	550	0.8413
	SBGMRES-DR(m)	**8125**	**10.542**	13420	21.555	29360	48.190
	BGMRES-S(m)	16456	32.017	**9025**	**17.780**	25699	52.065
	RB-SBGMRES-D(m)	**7785**	**10.274**	10652	17.950	31433	54.248
bidiag	**D-SBGMRES-DR(m, k)**	**460**	**0.6917**	460	0.6956	460	0.7041
	SBGMRES-DR(m)	**890**	**1.0425**	1020	1.1759	1125	1.2786
	BGMRES-S(m)	**781**	**1.2779**	879	1.3769	966	1.5821
	RB-SBGMRES-D(m)	**884**	**1.0528**	1007	1.2005	1084	1.2763

Here, $m = 20$, $k = 10$, $p = 5$, and $\epsilon_d = 1, 0.1, 0.01$. The best result in each row of the table is marked in bold

Example 5 The purpose of this example is to measure the performance of the proposed methods D-SBGMRES-DR and SBGMRES-DR compared with the other block Krylov subspace methods mentioned previously. This example relies on the four large test matrices Dubcova3, FEM_3D_thermal2, cage13 and cage14. The right-hand side matrix B is chosen in a way that the matrix $\texttt{ones}(n, p)$ is the exact solution of the linear systems of equations (1). The numerical computations are also carried out with $m = 30$ and $\epsilon_d = 1$. In addition, the values

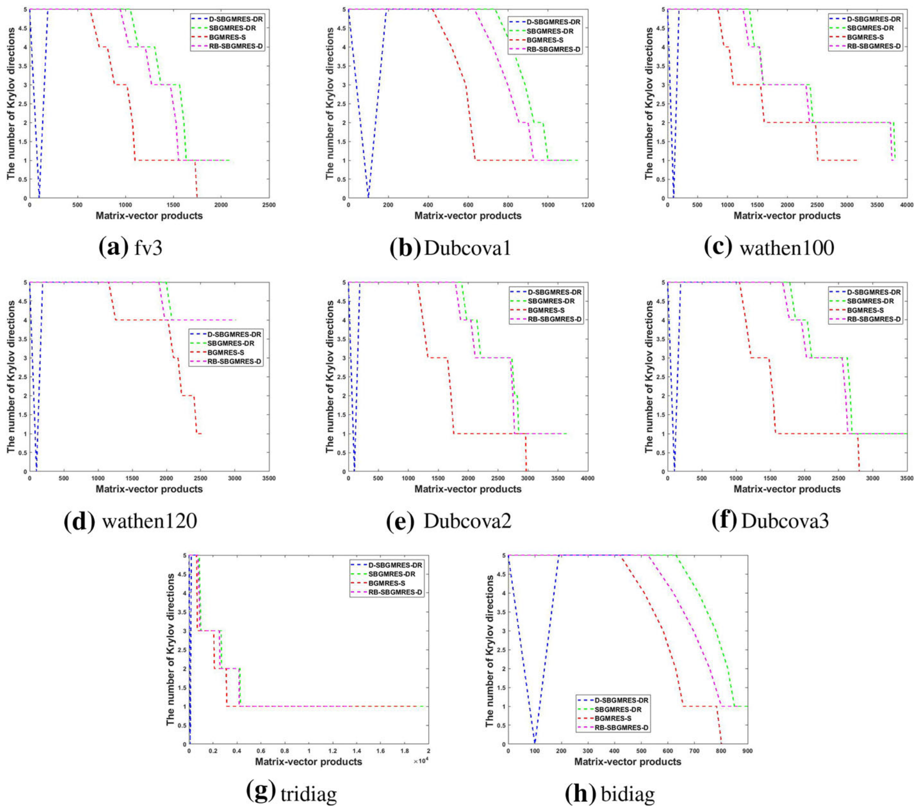


Fig. 2 Example 3: Illustration of the number of selected Krylov directions used in the D-SBGMRES-DR, SBGMRES-DR, BGMRES-S and RB-SBGMRES-D methods

of p and k are taken from $\{8, 10\}$. Table 8 lists the number of matrix–vector products, CPU time and relative residual norm for the different methods used in this example.

It is crystal clear from Table 8 that the D-SBGMRES-DR method succeeds excellently in achieving a remarkable level of efficiency among the other methods in terms of both the number of matrix–vector products and CPU time. Moreover, the reported numerical results disclose that the methods BGMRES-DR (for Dubcova3, FEM_3D_thermal2 and cage14), Ad-SBGMRES-DR (for FEM_3D_thermal2), Li-BGMRES and BGMRES (for cage14) fail to converge within 2000 restart cycles, while the D-SBGMRES-DR method enjoys a high performance in comparison with such methods.

Example 6 In this example, we appraise the applicability of our proposed methods and the other block Krylov subspace methods, utilized in our experiments, to the two-dimensional Helmholtz equation:

$$\begin{cases} -\Delta u - \beta^2 u = f, & \text{in } \Omega = [0, 1]^2, \\ u = 0, & \text{on the boundary } \partial\Omega, \end{cases} \tag{41}$$

in which the wavenumber β is fixed as $\beta = \pi$; for more information on the Helmholtz equations, we refer the reader to [14,28]. In order to discretize the equation (41), we make

Table 7 Example 4: performance results for the ten different block Krylov subspace methods

Method	fv3			Dubcova1		
	Mvps	CPU	res.norm	Mvps	CPU	res.norm
$m = 20$						
D-SBGMRES-DR(m, k)	**1340**	**4.686**	**1.6493e−07**	**960**	**5.914**	**1.8847e−07**
SBGMRES-DR(m)	2971	6.6411	9.9469e−07	1660	6.506	9.4136e−07
BGMRES-S(m)	2466	6.772	9.6341e−07	1320	7.414	9.7764e−07
RB-SBGMRES-D(m)	2942	9.350	7.0070e−07	1585	8.386	9.5893e−07
BGMRES-DR(m, k)	1800	5.733	6.5467e−07	1200	7.075	8.2244e−07
Ad-SBGMRES-DR(m, k)	1550	5.691	9.4611e−07	1040	6.005	8.6066e−07
Li-BGMRES(m)	8130	20.732	8.8408e−07	3950	21.636	6.4207e−07
SBCMRH(m)	13,690	49.632	3.4828e−06	6060	44.285	9.6146e−07
SBGMRES(m)	6490	17.599	9.9951e−07	3260	17.440	9.8695e−07
BGMRES(m)	7140	18.054	9.1293e−07	3780	22.557	5.9159e−07
Method	Tridiag			Bidiag		
	Mvps	CPU	res.norm	Mvps	CPU	res.norm
$m = 20$						
D-SBGMRES-DR(m, k)	**770**	**1.0264**	1.5614e−07	**770**	**0.4136**	**5.9382e−08**
SBGMRES-DR(m)	2440	2.6329	9.6934e−07	1360	0.5135	6.7182e−07
BGMRES-S(m)	11,463	20.000	9.9474e−07	1115	1.0281	9.3366e−07
RB-SBGMRES-D(m)	3964	4.8198	9.9100e−07	1275	0.5717	7.8459e−07
BGMRES-DR(m, k)	1200	1.1707	**2.0224e−08**	1000	0.4934	5.0039e−07
Ad-SBGMRES-DR(m, k)	830	1.2329	8.8446e−07	780	0.5931	9.9040e−07
Li-BGMRES(m)	109,990	70.027	6.9188e−04	3730	0.9922	5.1835e−07
SBCMRH(m)	5110	5.5807	2.3317e−06	1240	0.7627	1.3378e−06
SBGMRES(m)	2890	1.9995	9.9390e−07	1780	0.4234	9.2531e−07
BGMRE(m)	2520	2.0689	9.0429e−07	1890	0.4763	8.9447e−07
Method	fv3			Dubcova1		
	Mvps	CPU	res.norm	Mvps	CPU	res.norm
$m = 30$						
D-SBGMRES-DR(m, k)	**1160**	**7.8472**	3.0542e−07	**880**	**7.6293**	**4.7262e−07**
SBGMRES-DR(m)	2880	9.7863	5.1114e−07	1440	10.146	5.3194e−07
BGMRES-S(m)	2383	13.699	5.9056e−07	1091	11.619	8.7135e−07
RB-SBGMRES-D(m)	2774	12.312	5.3811e−07	1334	10.947	5.3354e−07
BGMRES-DR(m, k)	1800	9.4504	**1.4945e−07**	1500	12.850	8.8037e−09
Ad-SBGMRES-DR(m, k)	1510	10.070	**1.4945e−07**	960	9.8094	8.5030e−07
Li-BGMRES(m)	6070	17.003	5.7776e−07	3190	24.060	7.0469e−07
SBCMRH(m)	9280	40.494	4.8137e−06	5280	41.696	5.7355e−06
SBGMRES(m)	4530	17.755	9.7982e−07	2240	16.680	9.4658e−07
BGMRES(m)	4650	16.415	8.4275e−07	2480	21.897	7.4548e−07

Table 7 continued

Method	Tridiag			Bidiag		
	Mvps	CPU	res.norm	Mvps	CPU	res.norm
$m = 30$						
D-SBGMRES-DR(m, k)	**770**	**1.3815**	**1.3916e−08**	**590**	**0.4307**	8.1393e−07
SBGMRES-DR(m)	3460	4.0573	9.7662e−07	1170	0.6739	3.0967e−07
BGMRES-S(m)	3667	7.6632	9.9017e−07	940	1.7936	7.6827e−07
RB-SBGMRES-D(m)	1717	2.1446	8.5183e−07	1067	0.7288	4.9391e−07
BGMRES-DR(m, k)	1200	1.5743	3.7139e−08	910	0.6664	**3.5901e−09**
Ad-SBGMRES-DR(m, k)	770	1.3895	8.4561e−07	680	1.0012	8.5978e−07
Li-BGMRES(m)	99,510	76.709	9.8278e−07	2550	1.0053	8.6430e−07
SBCMRH(m)	18,030	21.660	2.7221e−06	690	0.5268	1.0573e−06
SBGMRES(m)	2400	1.7407	9.8309e−07	1290	0.7798	8.8628e−07
BGMRES(m)	2170	1.6098	6.2665e−07	1550	0.5573	1.4413e−07

Here, $m = 20, 30$, $k = 10$, $p = 10$ and $\epsilon_d = 1$. The best results are marked in bold

use of a classical second-order finite-difference method with the mesh sizes $h_x = \frac{1}{n_0+1}$ and $h_y = \frac{1}{n_1+1}$, where n_0 and n_1 are the number of points in the x and y directions, respectively. In this example, it is assumed that $n_0 = n_1$ in which the value of n_0 is taken as 150 and 256. In accordance with these settings, two nonsymmetric matrices with the dimensions $n = n_0^2 = 22500$ and $n = n_0^2 = 65536$ are generated.

Table 9 reports the number of restart cycles, the number of matrix–vector products and the CPU time for the different values of m and p. Here, the values of p and m are coming from $\{5, 10\}$ and $\{20, 30\}$, respectively. The parameters k and ϵ_d are also set to be 10 and 1, respectively. The right-hand side matrix B is also chosen as $B = \texttt{randn}(n, p)$.

It is found from Table 9 that the D-SBGMRES-DR, BGMRES-DR and Ad-SBGMRES-DR methods stand at a higher level of quality compared to the other methods. Moreover, D-SBGMRES-DR achieves the best values for the number of restart cycles, the number of matrix–vector products and the CPU time.

In order to conduct an illustrative investigation, Fig. 3 depicts the convergence histories of the ten block Krylov subspace methods for the case of $n = 22500$. We note here that the convergence curves of “Ad-SBGMRES-DR and BGMRES-DR” and “SBGMRES-DR, BGMRES-S, RB-SBGMRES-D, SBGMRES and BGMRES” are overlapped with each other. It is observed from this figure that the D-SBGMRES-DR, BGMRES-DR and Ad-SBGMRES-DR methods present the best results among the other methods in terms of both the number of restart cycles and the relative residual norm.

8 Conclusion

This paper presents two efficient methods derived from the simpler BGMRES (SBGMRES) method, which are called the simpler block GMRES method with vector deflation restarting (SBGMRES-DR) and the deflated simpler block GMRES method with vector deflation restarting (D-SBGMRES-DR). The SBGMRES-DR method is proposed to remove linearly or almost linearly dependent vectors created by the block Arnoldi process, while the D-

Table 8 Example 5: Performance results for the large test matrices Dubcova3, FEM_3D_thermal2, cage13 and cage14

Test matrix	Method	$p=k=8$			$p=k=10$		
		Mvps	CPU	res.norm	Mvps	CPU	res.norm
Dubcova3	**D-SBGMRES-DR(m,k)**	**204**	**15.590**	**2.3566e-07**	**233**	**18.033**	**3.3233e-07**
	SBGMRES-DR(m)	608	26.220	7.1245e-07	640	28.833	7.1245e-07
	BGMRES-S(m)	608	28.505	7.1245e-07	640	31.565	7.1245e-07
	RB-SBGMRES-D(m)	608	55.135	7.1245e-07	640	69.117	7.1245e-07
	BGMRES-DR(m,k)	–	–	–	–	–	–
	Ad-SBGMRES-DR(m,k)	1224	82.659	9.4172e-07	3360	238.65	9.8867e-07
	Li-BGMRES(m)	6904	493.69	8.3471e-09	8630	632.48	8.3471e-09
	SBCMRH(m)	⋆	⋆	⋆	⋆	⋆	⋆
	SBGMRES(m)	3776	261.96	9.9945e-07	4720	333.46	9.9881e-07
	BGMRES(m)	2976	218.00	7.1669e-07	3720	283.84	6.9738e-07
FEM_3D_thermal2	**D-SBGMRES-DR(m,k)**	**349**	**15.783**	7.0949e-07	**378**	**20.602**	7.6804e-07
	SBGMRES-DR(m)	418	16.777	7.3015e-07	440	21.567	7.3015e-07
	BGMRES-S(m)	418	17.741	7.3015e-07	440	30.003	8.9695e-06
	RB-SBGMRES-D(m)	418	35.678	7.3015e-07	440	46.726	7.3015e-07
	BGMRES-DR(m,k)	–	–	–	–	–	–
	Ad-SBGMRES-DR(m,k)	–	–	–	–	–	–
	Li-BGMRES(m)	5112	363.59	**6.4090e-09**	6390	461.14	**6.4090e-09**
	SBCMRH(m)	⋆	⋆	⋆	⋆	⋆	⋆
	SBGMRES(m)	2224	166.72	9.9896e-07	2760	209.68	9.8777e-07
	BGMRES(m)	2232	161.77	9.5530e-07	2790	209.81	8.5250e-07
cage13	**D-SBGMRES-DR(m,k)**	**24**	**4.0883**	4.4399e-10	**25**	**4.3330**	4.4399e-10
	SBGMRES-DR(m)	33	4.5144	4.4399e-10	35	4.5721	4.4399e-10
	BGMRES-S(m)	24	4.4562	1.6256e-07	26	4.4867	1.6256e-07
	RB-SBGMRES-D(m)	33	9.8579	4.4398e-10	35	12.310	4.4398e-10

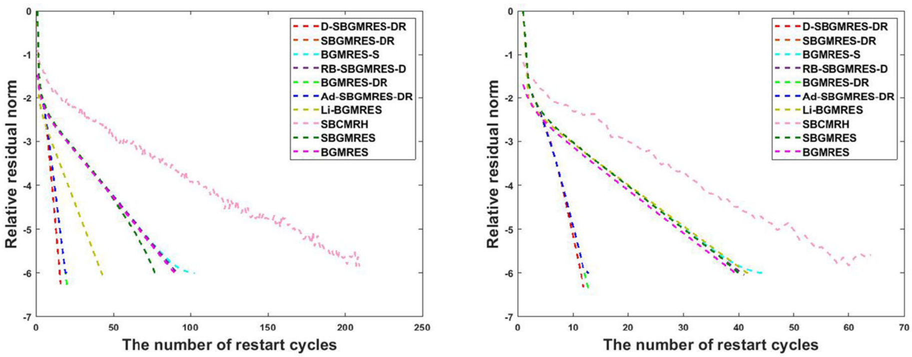
Table 8 continued

Test matrix	Method	$p = k = 8$			$p = k = 10$		
		Mvps	CPU	res.norm	Mvps	CPU	res.norm
	BGMRES-DR(m, k)	248	56.690	3.9494e-13	310	261.93	2.7632e-13
	Ad-SBGMRES-DR(m, k)	128	15.899	4.8778e-07	150	22.265	9.8814e-07
	Li-BGMRES(m)	504	117.39	**2.9526e-16**	630	378.73	**2.6760e-16**
	SBCMRH(m)	⋆	⋆	⋆	⋆	⋆	⋆
	SBGMRES(m)	128	17.169	9.7263e-07	150	27.422	7.7267e-07
	BGMRES(m)	248	56.069	3.9494e-13	310	75.405	2.7632e-13
cage14	**D-SBGMRES-DR**(m, k)	**23**	**11.140**	1.3283e-10	**23**	**11.039**	1.3283e-10
	SBGMRES-DR(m)	31	14.211	1.3283e-10	33	15.158	1.3283e-10
	BGMRES-S(m)	22	11.472	2.0604e-07	24	12.909	2.0604e-07
	RB-SBGMRES-D(m)	31	30.112	**1.3271e-10**	33	39.575	**1.3271e-10**
	BGMRES-DR(m, k)	†	†	†	†	†	†
	Ad-SBGMRES-DR(m, k)	104	790.76	8.6873e-07	130	61.691	6.7974e-07
	Li-BGMRES(m)	†	†	†	†	†	†
	SBCMRH(m)	⋆	⋆	⋆	⋆	⋆	⋆
	SBGMRES(m)	120	595.47	5.8546e-07	150	384.04	6.0044e-07
	BGMRES(m)	†	†	†	†	†	†

Table 9 Example 6: Performance results of each block Krylov subspace method on the two-dimensional Helmholtz equation

Method	$n = 22{,}500$			$n = 65{,}536$		
	Cycles	Mvps	CPU	Cycles	Mvps	CPU
$m = 20,\ p = 10$						
D-SBGMRES-DR(m, k)	**16**	**2860**	**30.602**	**34**	**6280**	**213.27**
SBGMRES-DR(m)	92	10,070	49.707	242	26,810	477.74
BGMRES-S(m)	103	7882	42.931	274	20,352	359.09
RB-SBGMRES-D(m)	92	9985	64.338	242	26,729	619.99
BGMRES-DR(m, k)	20	4000	32.730	48	9600	235.12
Ad-SBGMRES-DR(m, k)	20	3630	31.104	48	9370	214.31
Li-BGMRES(m)	42	13,750	143.85	240	53,010	1222.8
SBCMRH(m)	209	43,460	435.14	⋆	⋆	⋆
SBGMRES(m)	77	16,010	118.76	189	39,650	976.07
BGMRES(m)	90	18,900	136.35	239	50,190	1172.2
$m = 30,\ p = 5$						
D-SBGMRES-DR(m, k)	**12**	**1550**	**9.8060**	**23**	**3090**	**134.11**
SBGMRES-DR(m)	42	4580	20.665	113	11,990	255.27
BGMRES-S(m)	45	3565	17.751	124	9174	194.88
RB-SBGMRES-D(m)	41	4504	23.365	113	11,921	305.21
BGMRES-DR(m, k)	13	1890	12.436	26	3775	142.31
Ad-SBGMRES-DR(m, k)	13	1640	10.701	26	3560	139.08
Li-BGMRES(m)	41	6715	40.582	111	17,915	607.88
SBCMRH(m)	64	9250	128.96	⋆	⋆	⋆
SBGMRES(m)	39	6030	37.033	105	16,255	579.76
BGMRES(m)	40	6200	36.984	108	16,740	580.74

Here, $p = 5, 10$, $m = 20, 30$, $k = 10$ and $\epsilon_d = 1$. The best results are marked in bold, and the notation "⋆" shows that a method reaches the out-of-memory state



(a) $n = 22500,\ p = 10,\ m = 20.$ **(b)** $n = 22500,\ p = 5,\ m = 30.$

Fig. 3 Example 6: The convergence curves for the ten block Krylov subspace methods on the two-dimensional Helmholtz equation

SBGMRES-DR method employs the vector and eigenvalue deflation techniques at the same time. Compared with some BGMRES and SBGMRES methods based on the eigenvalue deflation or vector deflation, such as RB-SBGMRES-D, BGMRES-S, Ad-SBGMRES-DR and BGMRES-DR, the numerical studies point to the conclusion that the proposed D-SBGMRES-DR method, which integrates simultaneously both the eigenvalue and vector deflation techniques into the SBGMRES method, is highly effective and performs much better than the other methods.

Acknowledgements The authors would like to express their gratitude towards the anonymous reviewers for the valuable comments and suggestions that helped us to enhance the quality of this work. We are also grateful to Dr. Kathryn Lund for providing us with the codes of the loop-interchange block GMRES method.

Data Availibility Some or all data, models, or codes that support the findings of this study are available from the corresponding author upon reasonable request.

References

1. Abdaoui, l, Elbouyahyaoui, L., Heyouni, M.: The simpler block CMRH method for linear systems. Numer. Algorithms **84**, 1265–1293 (2020)
2. Al Daas, H., Grigori, L., Hénon, P., Ricoux, P.: Enlarged GMRES for solving linear systems with one or multiple right-hand sides. IMA J. Numer. Anal. **39**, 1924–1956 (2019)
3. Bloch, J.C., Breu, T., Frommer, A., Heybrock, S., Schaefer, K., Wettig, T.: Short-recurrence Krylov subspace methods for the overlap Dirac operator at nonzero chemical potential. Comput. Phys. Commun. **181**(8), 1378–1387 (2010)
4. Bloch, J.C., Heybrock, S.: A nested Krylov subspace method to compute the sign function of large complex matrices. Comput. Phys. Commun. **182**(4), 878–889 (2011)
5. Boojhawon, R., Bhuruth, M.: Restarted simpler GMRES augmented with harmonic ritz vectors. Future Gener. Comput. Syst. **20**(3), 389–397 (2004)
6. Bouyouli, R., Jbilou, K., Sadaka, R., Sadok, H.: Convergence properties of some block Krylov subspace methods for multiple linear systems. J. Comput. Appl. Math. **196**, 498–511 (2006)
7. Calandra, H., Gratton, S., Lago, R., Vasseur, X., Carvalho, L.M.: A modified block flexible GMRES method with deflation at each iteration for the solution of non-Hermitian linear systems with multiple right-hand sides. SIAM J. Sci. Comput. **35**(5), S345–S367 (2013)
8. Calandra, H., Gratton, S., Langou, J., Pinel, X., Vasseur, X.: Flexible variants of block restarted GMRES methods with application to geophysics. SIAM J. Sci. Comput. **34**(2), A714–A736 (2012)
9. Carpentieri, B., Jing, Y.-F., Huang, T.-Z.: The BiCOR and CORS iterative algorithms for solving nonsymmetric linear systems. SIAM J. Sci. Comput. **33**(5), 3020–3036 (2011)
10. Chen, G., Jia, Z.X.: Theoretical and numerical comparisons of GMRES and WZ-GMRES. Comput. Math. Appl. **47**(8–9), 1335–1350 (2004)
11. Clough, R.W., Penzien, J.: Structural Dynamics. McGrowHill Inc, New York (1975)
12. Davis, T.A., Hu, Y.: The University of Florida sparse matrix collection. ACM Trans. Math. Softw. (TOMS) **38**(1), 1 (2011)
13. Elbouyahyaoui, L., Heyouni, M., Tajaddini, A., Saberi-Movahed, F.: On restarted and deflated block FOM and GMRES methods for sequences of shifted linear systems. Numer. Algorithms **25**, 1–43 (2020)
14. Erlangga, Y.A., Vuik, C., Oosterlee, C.W.: On a class of preconditioners for solving the Helmholtz equation. Appl. Numer. Math. **50**, 409–425 (2004)
15. Freund, R.W.: Krylov-subspace methods for reduced-order modeling in circuit simulation. J. Comput. Appl. Math. **123**(1–2), 395–421 (2000)
16. Frommer, A., Lund, K., Szyld, D.B.: Block Krylov subspace methods for functions of matrices. Electron. Trans. Numer. Anal. **47**, 100–126 (2017)
17. Frommer, A., Lund, K., Szyld, D.B.: Block Krylov subspace methods for functions of matrices II: modified block FOM. SIAM J. Matrix Anal. Appl. **4**(2), 804–837 (2020)
18. Giraud, L., Gratton, S., Pinel, X., Vasseur, X.: Flexible GMRES with deflated restarting. SIAM J. Sci. Comput. **32**(4), 1858–1878 (2010)
19. Golub, G.H., Van Loan, C.F.: Matrix Computations, 4th edn. The Johns Hopkins University Press, Baltimore (2013)

20. Gu, G.-D., Cao, Z.-H.: A block GMRES method augmented with eigenvectors. Appl. Math. Comput. **121**(2–3), 271–289 (2001)
21. Gutknecht, M.H.: Block Krylov space methods for linear systems with multiple right-hand sides: an introduction. In: Siddiqi, A., Duff, I., Christensen, O. (eds.) Modern Mathematical Models, Methods and Algorithms for Real World Systems, pp. 420–447. Anamaya Publishers, New Delhi (2006)
22. Heyouni, M., Essai, A.: Matrix Krylov subspace methods for linear systems with multiple right-hand sides. Numer. Algorithms **40**, 137–156 (2005)
23. Jbilou, K., Messaoudi, A., Sadok, H.: Global FOM and GMRES algorithms for matrix equations. Appl. Numer. Math. **31**(1), 49–63 (1999)
24. Ji, H., Li, Y.-H.: A breakdown-free block conjugate gradient method. BIT Numer. Math. **57**, 379–403 (2017)
25. Jiránek, P., Rozložník, M.: Adaptive version of simpler GMRES. Numer. Algorithms **53**(1), 93–112 (2010)
26. J. Langou. Iterative Methods for Solving Linear Systems with Multiple Right-hand Sides. Ph.D. thesis, Ph. D. dissertation, INSA Toulouse (2003)
27. Liu, H., Zhong, B.: Simpler block GMRES for nonsymmetric systems with multiple right-hand sides. Electron. Trans. Numer. Anal. **30**, 1–9 (2008)
28. Meng, J., Zhu, P.-Y., Li, H.-B.: A block GCROT(m, k) method for linear systems with multiple right-hand sides. J. Comput. Appl. Math. **255**, 544–554 (2014)
29. 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)
30. Morgan, R.B.: Restarted block-GMRES with deflation of eigenvalues. Appl. Numer. Math. **54**(2), 222–236 (2005)
31. Rashedi, S., Ebadi, G., Birk, S., Frommer, A.: On short recurrence Krylov type methods for linear systems with many right-hand sides. J. Comput. Appl. Math. **300**, 18–29 (2016)
32. Robbé, M., Sadkane, M.: Exact and inexact breakdowns in the block GMRES method. Linear Algebra Appl. **419**(1), 265–285 (2006)
33. Saad, Y.: Iterative Methods for Sparse Linear Systems, vol. 82. SIAM, Philadelphia (2003)
34. 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)
35. Simoncini, V., Gallopoulos, E.: An iterative method for nonsymmetric systems with multiple right-hand sides. SIAM J. Sci. Comput. **16**(4), 917–933 (1995)
36. Simoncini, V., Gallopoulos, E.: A hybrid block GMRES method for nonsymmetric systems with multiple right-hand sides. J. Comput. Appl. Math. **66**(1–2), 457–469 (1996)
37. Soudais, P.: Iterative solution of a 3D scattering problem from arbitrary shaped multidielectric and multiconducting bodies. IEEE Trans. Antennas Propag. **42**(7), 954–959 (1994)
38. Sun, D.-L., Carpentieri, B., Huang, T.-Z., Jing, Y.-F.: A spectrally preconditioned and initially deflated variant of the restarted block GMRES method for solving multiple right-hand sides linear systems. Int. J. Mech. Sci. **144**, 775–787 (2018)
39. Sun, D.-L., Huang, T.-Z., Carpentieri, B., Jing, Y.-F.: A new shifted block GMRES method with inexact breakdowns for solving multi-shifted and multiple right-hand sides linear systems. J. Sci. Comput. **78**(2), 746–769 (2019)
40. Sun, D.-L., Huang, T.-Z., Carpentieri, B., Jing, Y.-F.: Flexible and deflated variants of the block shifted GMRES method. J. Comput. Appl. Math. **345**, 168–183 (2019)
41. Walker, H.F., Zhou, L.: A simpler GMRES. Numer. Linear Algebra Appl. **1**(6), 571–581 (1994)
42. Wu, Q., Bao, L., Lin, Y.: Residual-based simpler block GMRES for nonsymmetric linear systems with multiple right-hand sides. Adv. Math. Phys. **2018**, 1369707 (2018)
43. Xiang, Y.-F., Jing, Y.-F., Huang, T.-Z.: A new projected variant of the deflated block conjugate gradient method. J. Sci. Comput. **80**, 1116–1138 (2019)
44. Zhang, F.: Matrix Theory: Basic Results and Techniques. Springer, New York (2011)
45. Zhong, H.-X., Gu, X.-M.: A flexible and adaptive simpler GMRES with deflated restarting for shifted linear systems. Comput. Math. Appl. **78**(3), 997–1007 (2019)
46. Zhong, H.-X., Wu, G., Chen, G.-L.: A flexible and adaptive simpler block GMRES with deflated restarting for linear systems with multiple right-hand sides. J. Comput. Appl. Math. **282**, 139–156 (2015)