SIAM J. MATRIX ANAL. APPL.
Vol. 43, No. 2, pp. 710–739

A BLOCK MINIMUM RESIDUAL NORM SUBSPACE SOLVER WITH PARTIAL CONVERGENCE MANAGEMENT FOR SEQUENCES OF LINEAR SYSTEMS*

LUC GIRAUD†, YAN-FEI JING‡, AND YANFEI XIANG§

Abstract. We are concerned with the iterative solution of linear systems with multiple right-hand sides available one group after another with possibly slowly varying left-hand sides. For such sequences of linear systems, we first develop a new block minimum norm residual approach that combines two main ingredients. The first component exploits ideas from GCRO-DR [Parks et al., *SIAM J. Sci. Comput.,* 28 (2006), pp. 1651–1674], enabling us to recycle information from one solve to the next. The second component is the numerical mechanism for managing the partial convergence of the right-hand sides, referred to as inexact breakdown detection in IB-BGMRES [Robbé and Sadkane, *Linear Algebra Appl.,* 419 (2006), pp. 265–285], that enables the monitoring of the rank deficiency in the residual space basis expanded blockwise. Next, for the class of block minimum norm residual approaches that relies on a block Arnoldi-like equality between the search space and the residual space (e.g., any block GMRES or block GCRO variants), we introduce new search space expansion policies defined on novel criteria to detect the partial convergence. These novel detection criteria are tuned to the selected stopping criterion and targeted convergence threshold to best cope with the selected normwise backward error stopping criterion, enabling us to monitor the computational effort while ensuring the final accuracy of each individual solution. Numerical experiments are reported to illustrate the numerical and computational features of both the new block Krylov solvers and the new search space block expansion polices.

Key words. block subspace methods, augmentation, deflation, subspace recycling, partial convergence, inexact block rank deficiency, backward error stopping criterion

AMS subject classifications. 65F10, 65N22, 15A06

DOI. 10.1137/21M1401127

1. Introduction.

Many scientific and industrial simulations require the solution of a sequence of linear systems with multiple right-hand sides and possibly slowly changing left-hand sides. In that context, one has to solve a series of linear systems of the form

$$A^{(\ell)}X^{(\ell)} = B^{(\ell)}, \quad \ell = 1, 2, \ldots, \tag{1.1}$$

where, associated with the ℓth family, $A^{(\ell)} \in \mathbb{C}^{n\times n}$ is a square nonsingular matrix of large dimension n along the family index ℓ, $B^{(\ell)} = [b^{(\ell,1)}, b^{(\ell,2)}, \ldots, b^{(\ell,p^{(\ell)})}] \in \mathbb{C}^{n\times p^{(\ell)}}$ are simultaneously given right-hand sides of full rank with $p^{(\ell)} \ll n$, and $X^{(\ell)} = [x^{(\ell,1)}, x^{(\ell,2)}, \ldots, x^{(\ell,p^{(\ell)})}] \in \mathbb{C}^{n\times p^{(\ell)}}$ are the solutions to be computed. Both the coefficient matrix $A^{(\ell)}$ and right-hand sides $B^{(\ell)}$ change from one family to the next, and the families of linear systems are typically available in sequence.

*Received by the editors February 25, 2021; accepted for publication (in revised form) by U. M. Yang January 20, 2022; published electronically May 3, 2022.
https://doi.org/10.1137/21M1401127
Funding: The work of the second author was supported by National Natural Science Foundation of China grants 12071062 and 61772003, by Key Projects of Applied Basic Research in Sichuan Province grant 2020YJ0216, and by the Science Strength Promotion Programme of UESTC.

†HiePACS Project, Inria Bordeaux – Sud-Ouest, 33405 Talence Cedex, France (luc.giraud@inria.fr).
‡School of Mathematical Sciences, University of Electronic Science and Technology of China, Chengdu, Sichuan, 611731, China (yanfeijing@uestc.edu.cn).
§Cerfacs, Inria Bordeaux – Sud-Ouest, 33405 Talence Cedex, France (yanfei.xiang@inria.fr).

When solving sequences of linear systems such as (1.1), attractive approaches are those that can exploit information generated during the solution of a given system to accelerate the convergence for the next systems. Deflated restarting implements a similar idea between the cycles in the generalized minimum residual (GMRES) norm method [18, 20, 26]; it is realized by using a deflation subspace containing a few approximate eigenvectors deemed to hamper the convergence of the Krylov subspace methods [11, 12, 13]. Another alternative technique is the subspace recycling strategy proposed in the generalized conjugate residual with inner orthogonalization (GCRO) method and deflated restarting (GCRO-DR) method [16]. This latter method can reuse information accumulated in previous cycles as well as that accumulated during the solution of the previous families. Because the multiple right-hand sides of (1.1) are simultaneously available, block Krylov subspace methods are often considered as suitable candidates because of their capability of sharing search subspace that can be generated using basic linear algebra subprograms, such as level 3 BLAS-like implementation [10]. A common issue in block Krylov subspace methods is the rank deficiency that might appear during the expansion of the residual spaces, which is caused by the convergence of some individual solution or a linear combination of solution vectors. Such a rank deficiency problem could cause the block Arnoldi process to break down before the solutions for all the right-hand sides are found. For the sake of balancing robustness and convergence rate, Robbé and Sadkane proposed an inexact breakdown detection for the block GMRES algorithm (denoted by IB-BGMRES) [19], which could keep and reintroduce directions associated with the almost converged parts in next iteration if necessary. We refer the reader to [1, 2, 19], for relevant works on inexact breakdown detection, as well as to [22, 23, 24, 25, 27], for related variants of block Krylov subspace methods for solving linear systems with multiple right-hand sides.

The contribution of this paper is twofold. We first show how to combine subspace recycling techniques of GCRO-DR [16], for recycling spectral information at a new cycle/family, with the inexact breakdown detection introduced by Robbé and Sadkane in IB-BGMRES [19], for handling almost rank deficient blocks generated by the block Arnoldi procedure, to develop the IB-BGCRO-DR algorithm, a new recycling block GCRO-DR variant with partial convergence detection. This is a natural extension of our previous work on IB-BGMRES-DR [1] that enables the deflated restarting strategy proposed by Morgan [13] to be applied not only at restart but also when solving a sequence of linear systems. The IB-BGCRO-DR method can reuse spectral information from solutions in both the previous cycles and families, thus showing obvious advantages when solving sequences of linear systems like (1.1). In addition, we propose a flexible counterpart of the new algorithm which allows the use of a mixed arithmetic computation where all steps are computed with a selected working precision, except for the preconditioner which is performed with a reduced precision. The second contribution is related to the block search space expansion policies that can be further developed based on the partial convergence detection. In particular, for stopping criteria based on normwise backward error we introduce new strategies enabling one to focus on the computational effort while ensuring the final accuracy of each individual solution.

The remainder of this paper is organized as follows. Section 2 is devoted to the development of the new algorithm and contains some background that enables us to introduce the various numerical ingredients and notation required to design our algorithm. In section 2.1 we first recall the governing ideas of the minimum norm residual Krylov method GCRO in a single right-hand side setting, and in section 2.2

we briefly present its block variant. Next, in section 2.3 we present how the original inexact breakdown detection mechanism [19] introduced for block GMRES can be applied to block GCRO as well. These two main ingredients are combined to develop the new algorithm IB-BGCRO-DR in section 2.4 and its flexible preconditioning variant, referred to as IB-BFGCRO-DR, in section 2.5. In section 3, we describe how to extend the original inexact breakdown detection mechanism to best adapt the computational effort and reach the targeted accuracy prescribed by the stopping criterion defined in terms of normwise backward errors for the individual solutions. In particular, we derive strategies for managing the situation where the different right-hand sides need to be solved with different convergence thresholds. We also present policies adapted to a stopping criterion based on normwise backward error on the right-hand side only (i.e., classical residual norm scaled by the norm of the right-hand side) or the more general one used to establish the backward stability of GMRES [14]. Section 4 presents some detailed remarks on computational and algorithmic aspects; the associated pseudocode of the IB-BGCRO-DR algorithm is presented as well. In section 5 we present numerical experiments that illustrate the benefits of the new algorithm with both constant and slowly varying successive linear systems with multiple right-hand sides, and we introduce as well the numerical capabilities of the novel search space expansion policies. Finally, we conclude with some detailed remarks in section 6.

The symbol $||\cdot||$ denotes the Euclidean norm default for both vectors and matrices, and the Frobenius norm is denoted with the subscript $_F$. The superscript ^{H} denotes the transpose conjugate and ^{T} stands for transpose. Because much notation is involved, we make certain choices to improve the readability of the paper. The vectors are described by lowercase letters; matrices with multiple columns are described by uppercase letters; calligraphy uppercase letters, e.g., $\mathscr{V}$, represent matrices whose columns are enlarged by multiple columns at each iteration (as commonly appear in the block Krylov context); and uppercase blackboard bold letters, e.g., $\mathbb{V}$, refer to the block Krylov basis generated at each iteration. The superscript † refers to the Moore–Penrose inverse. For convenience of the algorithm illustration and presentation, some MATLAB notation is used. Without special note, a subscript $_j$ for a vector (in the single right-hand case) or a matrix (in the block case) is used to indicate that the vector or matrix is obtained at iteration j, and a positive subscript integer $_m$ represents the maximal iteration number of each (block) Krylov cycle. All the involved recycling subspaces of dimension k are described as a matrix with the subscript $_k$, whose columns form a basis. A matrix $C \in \mathbb{C}^{m\times\ell}$ consisting of m rows and ℓ columns sometimes is denoted as $C_{m\times\ell}$ explicitly. The identity and null matrices of dimension m are denoted, respectively, by I_m and 0_m, or by just I and 0 when the dimension is evident from the context. For a matrix $C \in \mathbb{C}^{m\times\ell}$, the singular values of C are denoted by $\sigma_1(C) \geq \cdots \geq \sigma_{min(m,\ell)}(C)$ in descending order; furthermore, we denote by $\text{span}(C)$ the space spanned by the columns of C.

For simplicity and notational convenience, in the rest of this paper we drop the superscript $^{(\ell)}$ in $B^{(\ell)}$ and $X^{(\ell)}$ whenever we consider solving the current ℓth family of linear systems in the entire sequence of families. We indicate the superscript for a family order explicitly when necessary. That is, suppose that the current ℓth family of linear systems to be solved is

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

where, $A \in \mathbb{C}^{n\times n}$ is the current square nonsingular matrix of dimension n, $B = [b^{(1)}, b^{(2)}, \ldots, b^{(p)}] \in \mathbb{C}^{n\times p}$ are the right-hand sides given simultaneously, and $X = [x^{(1)}, x^{(2)}, \ldots, x^{(p)}] \in \mathbb{C}^{n\times p}$ are the solutions to be computed.

2. Block GCRO-DR with partial convergence detection. For the sake of completeness, this section contains some (possibly well-known) background which enables us to introduce the notation required to describe the new algorithm and detail its properties. In that respect, we first recall the main ingredients of the subspace recycling techniques existing in the minimum residual Krylov methods GCRO [7] and GCRO-DR [16] that are presented in the single right-hand side context. Next, we introduce the straightforward extension to the multiple right-hand sides framework that is the block formulation of GCRO-DR (BGCRO-DR) [15, 17]. Then the driving ideas of partial convergence detection [19], along with the corresponding block Arnoldi-like recurrence equation, are derived in the block GCRO-DR context, leading to the new IB-BGCRO-DR algorithm.

2.1. GCRO. The background of GCRO [7] is briefly reviewed in the case of a single right-hand side and then extended to the block case. The GCRO method relies on a given full-rank matrix $U_k \in \mathbb{C}^{n\times k}$ and on a matrix C_k as the image of U_k by A satisfying the relations

$$AU_k = C_k, \tag{2.1}$$

$$C_k^H C_k = I_k. \tag{2.2}$$

For the solution of a single right-hand side linear system $Ax = b$ and a given initial guess x_0, the governing idea is to first define $x_1 \in x_0 + \mathrm{Range}(U_k)$ that minimizes the residual norm. From x_1 and its associated residual r_1, Arnoldi iterations are performed to enlarge the nested orthonormal basis of the residual spaces. The vector

$$x_1 = \underset{x\in x_0+\mathrm{Range}(U_k)}{\operatorname{argmin}} ||b - Ax||$$

is defined by

$$x_1 = x_0 + U_k C_k^H r_0, \text{ and } r_1 = (I - C_k C_k^H) r_0 \text{ such that } r_1 \in C_k^{\perp}.$$

Starting from the unit vector $v_1 = r_1/\|r_1\|$, the Arnoldi procedure enables us to form an orthonormal basis $V_m = [v_1, \ldots, v_m]$ of the Krylov space $\mathcal{K}_m((I - C_k C_k^H)A, v_1) = \mathrm{span}(v_1, (I - C_k C_k^H)Av_1, \ldots, ((I - C_k C_k^H)A)^{m-1} v_1)$, yielding an Arnoldi-like relation in matrix form as

$$(I - C_k C_k^H)AV_m = V_{m+1}\underline{H}_m, \tag{2.3}$$

where the top square part of $\underline{H}_m \in \mathbb{C}^{(m+1)\times m}$ is upper Hessenberg, and only the last entry of its last row is nonzero. Combining (2.1) and (2.3) into one matrix form allows us to write a relation quite similar to an Arnoldi equality that reads

$$A\widehat{W}_m = \widehat{V}_{m+1}\underline{G}_m,$$

where the columns of $\widehat{W}_m = [U_k, V_m]$ define a basis of the search space, columns of $\widehat{V}_{m+1} = [C_k, V_{m+1}]$ make up an orthonormal basis of the residual space, and $\underline{G}_m = \begin{bmatrix} I_k & B_m \\ 0_{(m+1)\times k} & \underline{H}_m \end{bmatrix} \in \mathbb{C}^{(k+m+1)\times(k+m)}$, with $\widehat{V}_{m+1}^H \widehat{V}_{m+1} = I_{m+1}$ and $B_m = C_k^H A V_m$. The minimum residual norm solution in the affine space $x_1 + \mathrm{Range}(\widehat{W}_m)$ can be written as $x_m = x_1 + \widehat{W}_m y_m$, where

$$y_m = \underset{y\in\mathbb{C}^{k+m}}{\operatorname{argmin}} \|c - \underline{G}_m y\|,$$

and $c = \widehat{V}_{m+1}^H r_1 = (0_k, \|r_1\|, 0_m)^T \in \mathbb{C}^{k+m+1}$ are the components of the residual associated with x_1 in the residual space spanned by the columns of $\widehat{V}_{m+1}$.

GCRO and GMRES [20] both belong to the family of residual norm minimization approaches and rely on an orthonormal basis of the residual space. In addition to sharing the Arnoldi procedure to form part or all of this basis, they also share the property of "happy breakdown"; that is, if the search space cannot be enlarged because the new direction computed by the Arnoldi process is the null vector, then the solution is exactly found in the search space. This sharing of features extends to the block context for the solution of linear systems with multiple right-hand sides; in particular, the inexact breakdown principle introduced in [19] in the context of block GMRES can be extended to block GCRO, as discussed in what follows. The purpose of the partial convergence detection is to prevent, in an elegant and effective way, the loss of numerical rank of the search space basis; this turns out to be also a way to monitor the search space expansion according to the final target accuracy.

2.2. Block GCRO. The straightforward extension of the GCRO method in the block context is briefly described below. To facilitate reading, we do not use the calligraphic form in the notation but keep the same letters to denote the block counterparts of the quantities involved in the method. Starting from the block initial guess $X_0 = [x_0^{(1)}, x_0^{(2)}, \ldots, x_0^{(p)}] \in \mathbb{C}^{n\times p}$ and associated initial residual block $R_0 = B - AX_0$, one can define

$$X_1 = \underset{X\in X_0+\mathrm{Range}(U_k)}{\operatorname{argmin}} ||B - AX||_F,$$

given by

$$X_1 = X_0 + U_k C_k^H R_0, \text{ and } R_1 = (I - C_k C_k^H)R_0 \text{ such that } R_1 \in C_k^{\perp}. \tag{2.4}$$

For the sake of simplicity, we first assume that R_1 is of full rank and denote $R_1 = \mathbb{V}_1\Lambda_1$ as its reduced QR-factorization. The orthonormal block $\mathbb{V}_1$ is then used to build the search space via m steps of the block Arnoldi procedure, depicted in Algorithm 1, to generate $\mathscr{V}_m = [\mathbb{V}_1, \ldots, \mathbb{V}_m]$, whose columns form an orthonormal basis of $\mathcal{K}_m((I - C_kC_k^H)A, \mathbb{V}_1) = \bigoplus_{t=1}^p \mathcal{K}_m((I - C_kC_k^H)A, v_1^{(t)})$.

Algorithm 1. *Block Arnoldi procedure with deflation of the C_k space.*

Given a nonsingular coefficient matrix $A \in \mathbb{C}^{n\times n}$, choose a matrix $\mathbb{V}_1 \in \mathbb{C}^{n\times p}$ with orthonormal columns
for $j = 1, 2, \ldots, m$ **do**
  Compute $W_j = (I - C_kC_k^H)A\mathbb{V}_j$
  for $i = 1, 2, \ldots, j$ **do**
    $H_{i,j} = \mathbb{V}_i^H W_j$
    $W_j = W_j - \mathbb{V}_i H_{i,j}$
  end for
  $W_j = \mathbb{V}_{j+1}H_{j+1,j}$ (reduced QR-factorization of W_j)
end for

The block Arnoldi procedure leads to the matrix equality

$$(I - C_kC_k^H)A\mathscr{V}_m = \mathscr{V}_{m+1}\underline{\mathscr{H}}_m, \tag{2.5}$$

where $\underline{\mathscr{H}}_m$ is a block Hessenberg matrix with (i,j) block defined by $H_{i,j}$. Similarly to the single right-hand side case, (2.1) and (2.5) can be gathered into matrix form

$$A\widehat{\mathscr{W}_m} = \widehat{\mathscr{V}}_{m+1}\underline{\mathcal{G}}_m, \tag{2.6}$$

where $\widehat{\mathscr{W}_m} = [U_k, \mathscr{V}_m] \in \mathbb{C}^{n\times(k+mp)}$, $\widehat{\mathscr{V}}_{m+1} = [C_k, \mathscr{V}_{m+1}] \in \mathbb{C}^{n\times(k+(m+1)p)}$, and $\underline{\mathcal{G}}_m = \begin{bmatrix} I_k & \mathcal{B}_m \\ 0_{(m+1)p\times k} & \underline{\mathscr{H}}_m \end{bmatrix} = \begin{bmatrix} & & c\mathcal{G}_m \\ & & \\ 0_{p\times(k+(m-1)p)} & & H_{m+1,m} \end{bmatrix} \in \mathbb{C}^{(k+(m+1)p)\times(k+mp)}$ with $\widehat{\mathscr{V}}_{m+1}^H\widehat{\mathscr{V}}_{m+1} = I_{k+(m+1)p}$ and $\mathcal{B}_m = C_k^H A\mathscr{V}_m \in \mathbb{C}^{k\times mp}$; here $mp = m \times p$. The minimum residual norm solution in the affine space $X_1 + \mathrm{Range}(\widehat{W}_m)$ can be written as $X_m = X_1 + \widehat{\mathscr{W}_m}Y_m$, where

$$Y_m = \underset{Y\in\mathbb{C}^{(k+mp)\times p}}{\operatorname{argmin}} \|\mathcal{C} - \underline{\mathcal{G}}_m Y\|_F,$$

$\mathcal{C} = \widehat{\mathscr{V}}_{m+1}^H R_1 = (0_{p\times k}, \Lambda_1^T, 0_{p\times mp})^T \in \mathbb{C}^{(k+(m+1)p)\times p}$, and the columns of $\mathcal{C}$ are the components of the initial residual block R_1 in the residual space $\widehat{\mathscr{V}}_{m+1}$.

2.3. Block GCRO with partial convergence detection. When one solution or a linear combination of solutions has converged, the block Arnoldi procedure implemented to build an orthonormal basis of $\mathcal{K}_j((I-C_kC_k^H)A, \mathbb{V}_1)$ needs to be modified to account for this partial convergence. This partial convergence is characterized by a numerical rank deficiency in the new p directions that are usually introduced for enlarging the search space at the next iteration. In [19], the authors present an elegant numerical variant that enables the detection of what is referred to as inexact breakdowns. In that approach the directions that have a low contribution to the residual block are discarded from the candidate set of vectors used to expand the search space at the next iteration, but these directions are reintroduced in iterations afterward if necessary. In this section, we try to give an insight and the main equality required to derive the IB-BGCRO-DR algorithm. We refer the reader to the original paper [19] for a detailed and complete description. For the sake of simplicity and easy cross reference, we adopt most of the notation from [1, 19].

When a partial convergence occurs, not all the space spanned by W_j is considered to build $\mathbb{V}_{j+1}$ in order to expand the search space. For the sake of simplicity, we assume that $p_1 = p$, and we denote by p_{j+1} the number of columns of the block orthonormal basis vector $\mathbb{V}_{j+1}$. Then $\mathbb{V}_{j+1} \in \mathbb{C}^{n\times p_{j+1}}, W_j \in \mathbb{C}^{n\times p_j}$, and $H_{j+1,j} \in \mathbb{C}^{p_{j+1}\times p_j}$. As a consequence the dimension of the search space $\mathcal{K}_j((I - C_kC_k^H)A, \mathbb{V}_1)$ considered at the jth iteration is no longer necessarily equal to $j \times p$ but is equal to $n_j = \sum_{i=1}^{j} p_i$, that is, the sum of the column rank of $\mathbb{V}_i's$ $(i = 1, \ldots, j)$.

When no partial convergence has occurred, that is, $p_{j+1} = p_j = \cdots = p_1 = p$, the range of W_j has always been used to enlarge the search space, and we obtain the block relation given by (2.6). To account for a numerical deficiency in the residual block $R_j = B - AX_j$ in a way that is described later, Robbé and Sadkane [19] proposed splitting

$$W_j = \mathbb{V}_{j+1}H_{j+1,j} + Q_j \tag{2.7}$$

such that the columns of Q_j and $\mathbb{V}_{j+1}$ are orthogonal to each other and only $\mathbb{V}_{j+1}$ is used to enlarge $\mathscr{V}_j$ to form $\mathscr{V}_{j+1}$. We can then extend (2.6) into

$$A\widehat{\mathscr{W}_j} = \widehat{\mathscr{V}_j}\mathcal{G}_j + [0_{n\times k},\ \mathcal{Q}_{j-1},\ W_j], \tag{2.8}$$

where $\mathcal{G}_j \in \mathbb{C}^{(k+n_j)\times(k+n_j)}$ is the first $k+n_j$ rows of $\underline{\mathcal{G}}_j \in \mathbb{C}^{(k+n_j+p)\times(k+n_j)}$, and $\mathcal{Q}_{j-1} = [Q_1, \ldots, Q_{j-1}] \in \mathbb{C}^{n\times n_{j-1}}$ accounts for all the discarded directions. The matrix $\mathcal{Q}_{j-1}$ is rank deficient, and it reduces to the zero matrix of $\mathbb{C}^{n\times n_{j-1}}$ as long as no partial convergence has occurred.

In order to characterize a minimum norm solution in the space spanned by $\widehat{\mathscr{W}_j}$ using (2.8) we need to form an orthonormal basis of the space spanned by $[\widehat{\mathscr{V}_j}, \mathcal{Q}_{j-1}, W_j]$. This is performed by first orthogonalizing $\mathcal{Q}_{j-1}$ against $\widehat{\mathscr{V}_j}$, that is, $\widetilde{\mathcal{Q}}_{j-1} = (I - \widehat{\mathscr{V}_j}\widehat{\mathscr{V}_j}^H)\mathcal{Q}_{j-1}$. Because $\mathcal{Q}_{j-1}$ is of rank deficiency, so is $\widetilde{\mathcal{Q}}_{j-1}$, which can be written as

(2.9)
$$\widetilde{\mathcal{Q}}_{j-1} = P_{j-1}\mathbb{G}_{j-1} \text{ with } \begin{cases} P_{j-1} \in \mathbb{C}^{n\times q_j} \text{ has orthonormal columns with } \widehat{\mathscr{V}_j}^H P_{j-1} = 0, \\ \mathbb{G}_{j-1} \in \mathbb{C}^{q_j\times n_{j-1}} \text{ is of full rank with } q_j = p - p_j. \end{cases}$$

Next, W_j that is already orthogonal to $\widehat{\mathscr{V}_j}$ is made to be orthogonal to P_{j-1} with $W_j - P_{j-1}E_j$ where $E_j = P_{j-1}^H W_j$; then one computes $\widetilde{W}_j D_j$ with $\widetilde{W}_j \in \mathbb{C}^{n\times p_j}$ and $D_j \in \mathbb{C}^{p_j\times p_j}$ by carrying out the reduced QR-factorization of the tall and skinny matrix $W_j - P_{j-1}E_j$. Eventually, the columns of the matrix $[\widehat{\mathscr{V}_j}, P_{j-1}, \widetilde{W}_j]$ form an orthonormal basis of the residual space spanned by $[\widehat{\mathscr{V}_j}, \mathcal{Q}_{j-1}, W_j]$.

With this new basis, (2.8) reads

$$\begin{aligned} A[U_k, \mathscr{V}_j] =& [C_k, \mathscr{V}_j] \begin{bmatrix} I & \mathcal{B}_j \\ 0 & \mathscr{L}_j \end{bmatrix} + \left[0_k, P_{j-1}\mathbb{G}_{j-1}, \left[P_{j-1}, \widetilde{W}_j\right] \begin{bmatrix} E_j \\ D_j \end{bmatrix}\right] \\ =& \left[C_k, \mathscr{V}_j, [P_{j-1}, \widetilde{W}_j]\right] \begin{bmatrix} I_k & & \mathcal{B}_j \\ & & \mathscr{L}_j \\ 0_{(n_j+p)\times k} & \mathbb{G}_{j-1} & E_j \\ & 0 & D_j \end{bmatrix}, \end{aligned} \tag{2.10}$$

where

$$\mathscr{L}_j = \begin{bmatrix} H_{1,1} & H_{1,2} & H_{1,3} & \cdots & H_{1,j} \\ H_{2,1} & H_{2,2} & H_{2,3} & \cdots & H_{2,j} \\ \mathbb{V}_3^H Q_1 & H_{3,2} & H_{3,3} & \cdots & H_{3,j} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbb{V}_j^H Q_1 & \cdots & \mathbb{V}_j^H Q_{j-2} & H_{j,j-1} & H_{j,j} \end{bmatrix} \in \mathbb{C}^{n_j\times n_j}$$

is no longer upper Hessenberg as soon as one partial convergence occurs, i.e., $\exists \ell$ s.t. $Q_\ell \neq 0$.

Equation (2.10) can be rewritten in a more compact form as

$$A[U_k, \mathscr{V}_j] = \left[C_k, \mathscr{V}_j, [P_{j-1}, \widetilde{W}_j]\right] \underline{\mathscr{F}}_j,$$

so that the least squares problem to be solved to compute the minimum residual norm solution associated with the generalized Arnoldi relation (2.10) becomes

$$Y_j = \underset{Y\in\mathbb{C}^{(k+n_j)\times p}}{\operatorname{argmin}} \left\|\Lambda_j - \underline{\mathscr{F}}_j Y\right\|_F, \tag{2.11}$$

with

$$\underline{\mathscr{F}}_j = \begin{bmatrix} I_k & & \mathcal{B}_j \\ & & \mathscr{L}_j \\ 0_{(n_j+p)\times k} & \mathbb{G}_{j-1} & E_j \\ & 0 & D_j \end{bmatrix} = \begin{bmatrix} \mathscr{F}_j \\ \mathbb{H}_j \end{bmatrix} \in \mathbb{C}^{(k+n_j+p)\times(k+n_j)} \tag{2.12}$$

and $\Lambda_j = \left[\begin{smallmatrix} 0_{k\times p} \\ \Lambda_1 \\ 0_{n_j\times p}\end{smallmatrix}\right] \in \mathbb{C}^{(k+n_j+p)\times p}$, where $\mathscr{F}_j = \left[\begin{smallmatrix} I_k & \mathcal{B}_j \\ 0_{n_j\times k} & \mathscr{L}_j\end{smallmatrix}\right] \in \mathbb{C}^{(k+n_j)\times(k+n_j)}$ and $\mathbb{H}_j = [0_{p\times k} \begin{smallmatrix}\mathbb{G}_{j-1} & E_j \\ 0 & D_j\end{smallmatrix}] \in \mathbb{C}^{p\times(k+n_j)}$.

The numerical mechanism for selecting $\mathbb{V}_{j+1}$ out of $[P_{j-1}, \widetilde{W}_j]$ follows the same ideas as discussed in [1, 19] in the context of block GMRES. The governing idea consists of building an orthonormal basis for the directions that contribute the most to the individual residual norms and make them larger than a prescribed threshold τ.

Specifically, the singular value decomposition (SVD) is applied to the least squares residuals

$$\Lambda_j - \underline{\mathscr{F}}_j Y_j = \mathbb{U}_{1,L}\Sigma_1\mathbb{U}_{1,R}^H + \mathbb{U}_{2,L}\Sigma_2\mathbb{U}_{2,R}^H, \tag{2.13}$$

where Σ_1 contains the p_{j+1} singular values greater than or equal to the prescribed threshold τ. Then we decompose $\mathbb{U}_{1,L} = \left(\begin{smallmatrix}\mathbb{U}_1^{(1)} \\ \mathbb{U}_1^{(2)}{}_x\end{smallmatrix}\right)$ in accordance with $[[C_k, \mathscr{V}_j], [P_{j-1}, \widetilde{W}_j]]$, that is, $\mathbb{U}_1^{(1)} \in \mathbb{C}^{(k+n_j)\times p_{j+1}}$ and $\mathbb{U}_1^{(2)} \in \mathbb{C}^{p\times p_{j+1}}$. Because the objective is to construct an orthonormal basis, we consider a unitary matrix $[\mathbb{W}_1, \mathbb{W}_2]$ such that $\mathrm{Range}(\mathbb{W}_1) = \mathrm{Range}(\mathbb{U}_1^{(2)})$. The new set of orthonormal candidate vectors used to expand the search space

$$\mathbb{V}_{j+1} = \left[P_{j-1}, \widetilde{W}_j\right] \mathbb{W}_1 \tag{2.14}$$

is the set that contributes the most to the residual norms, while

$$P_j = \left[P_{j-1}, \widetilde{W}_j\right] \mathbb{W}_2$$

is the new set of discarded directions with orthonormal columns. Through this mechanism, directions that have been discarded at a given iteration can be reintroduced if the residual block has a large component along them. Furthermore, this selection strategy ensures that all the solutions have converged when p partial convergence has been detected. We do not give details of the calculation but instead refer the reader to section 3 of [19] for a complete description; we only state that via this decomposition, the main terms that appear in (2.10) can be computed incrementally.

2.4. Subspace recycling policies along with partial convergence detection. So far, we have not made any specific assumption about the definition of the recycling space U_k except that it has full column rank. In the context of subspace recycling, one key point is to specify what subspace is to be recycled at restart. At the cost of the extra storage of k vectors, block GCRO offers more flexibility than block GMRES in the choice of the recycling space. This extra storage, which enables us to remove the constraints so that the search space is included in the residual space, allows us to consider any subspace to be deflated at restart. In particular, either of the two classical alternatives, the Rayleigh–Ritz procedure or the harmonic-Ritz procedure, can be considered to compute the targeted approximated eigenvectors to define U_k and C_k at restart. To keep this paper to a reasonable length, we present details on building a recycling subspace based on the harmonic-Ritz projection. We refer the reader to our technical report [9, sections 2.4 and 5.1] for corresponding discussions on the implementation based on the Rayleigh–Ritz procedure.

DEFINITION 1 (harmonic-Ritz projection). *Consider a subspace $\mathcal{W}$ of $\mathbb{C}^n$. Given a general nonsingular matrix $A \in \mathbb{C}^{n\times n}$, $\lambda \in \mathbb{C}$, and $g \in \mathcal{W}$, we see that (λ, g) is a*

harmonic-Ritz pair of A with respect to the space $\mathcal{W}$ if and only if

$$Ag - \lambda g \perp A\mathcal{W}$$

or, equivalently,

$$\forall w \in \mathrm{Range}(A\mathcal{W}), \quad w^H (Ag - \lambda g) = 0.$$

The vector g is a harmonic-Ritz vector associated with the harmonic-Ritz value λ.

Once the maximum size of the search space has been reached, we have

$$A\widehat{\mathscr{W}}_m = \widehat{\mathscr{V}}_{m+1}\underline{\mathscr{F}}_m = \left[C_k, \mathscr{V}_m, [P_{m-1}, \widetilde{W}_m]\right] \underline{\mathscr{F}}_m, \tag{2.15}$$

$$X_m = X_1 + \widehat{\mathscr{W}}_m Y_m, \tag{2.16}$$

$$R_m = B - AX_m = \left[C_k, \mathscr{V}_m, [P_{m-1}, \widetilde{W}_m]\right] (\Lambda_m - \underline{\mathscr{F}}_m Y_m), \tag{2.17}$$

$$Y_m = \underset{Y \in \mathbb{C}^{(k+n_m)\times p}}{\mathrm{argmin}} \|\Lambda_m - \underline{\mathscr{F}}_m Y\|_F, \quad \Lambda_m = [0_{p\times k}, \Lambda_1^T, 0_{p\times n_m}]^T. \tag{2.18}$$

Then, a restart procedure has to be implemented to possibly refine the spectral information to be recycled during the next cycle. Based on these equalities we will compute the approximated eigen-information as shown in Proposition 1 and then use it to define the new deflation basis U_k^{new} and its orthonormal image C_k^{new} by A as described in Theorem 1.

PROPOSITION 1. *At restart of IB-BGCRO-DR, the update of the recycling subspace for the next cycle relies on the computation of harmonic-Ritz vectors $\widehat{\mathscr{W}}_m g_i \in \mathrm{span}(\widehat{\mathscr{W}}_m)$ of A with respect to $\widehat{\mathscr{W}}_m = [U_k, \mathscr{V}_m] \in \mathbb{C}^{n\times(k+n_m)}$.*

The harmonic-Ritz pairs $(\theta_i, \widehat{\mathscr{W}}_m g_i)$ to be possibly used for the next restart satisfy

$$\underline{\mathscr{F}}_m^H \underline{\mathscr{F}}_m g_i = \theta_j \underline{\mathscr{F}}_m^H \widehat{\mathscr{V}}_{m+1}^H \widehat{\mathscr{W}}_m g_i \quad \textit{for } 1 \le i \le k + n_m, \tag{2.19}$$

where

$$\widehat{\mathscr{V}}_{m+1}^H \widehat{\mathscr{W}}_m = \begin{bmatrix} C_k^H U_k & 0_{k\times n_m} \\ \mathscr{V}_m^H U_k & I_{n_m} \\ P_{m-1}^H U_k & \\ \widetilde{W}_m^H U_k & 0_{p\times n_m} \end{bmatrix} \in \mathbb{C}^{(k+n_m+p)\times(k+n_m)}.$$

Proof. The proofs basically rely on some matrix computations as shortly described below.

According to Definition 1, each harmonic-Ritz pair $(\theta_i, \widehat{\mathscr{W}}_m g_i)$ satisfies

$$\forall w \in \mathrm{Range}(A\widehat{\mathscr{W}}_m), \quad w^H (A\widehat{\mathscr{W}}_m g_i - \theta_i \widehat{\mathscr{W}}_m g_i) = 0,$$

which is equivalent to

$$(A\widehat{\mathscr{W}}_m)^H (A\widehat{\mathscr{W}}_m g_i - \theta_i \widehat{\mathscr{W}}_m g_i) = 0.$$

Substituting (2.15) into the above leads to

$$\left(\widehat{\mathscr{V}}_{m+1}\underline{\mathscr{F}}_m\right)^H \left(\widehat{\mathscr{V}}_{m+1}\underline{\mathscr{F}}_m g_i - \theta_i \widehat{\mathscr{W}}_m g_i\right) = 0. \tag{2.20}$$

Because the columns of $\widehat{\mathscr{V}}_{m+1} = [C_k, \mathscr{V}_m, [P_{m-1}, \widetilde{W}_m]]$ generated at the end of each cycle are orthonormal, (2.20) becomes

$$\underline{\mathscr{F}}_m^H \underline{\mathscr{F}}_m g_i - \theta_i \underline{\mathscr{F}}_m^H \widehat{\mathscr{V}}_{m+1}^H \widehat{\mathscr{W}}_m g_i = 0,$$

which gives the formulation (2.19). ∎

Depending on the region of the spectrum that is intended to be deflated (e.g., subspace associated with the smallest and/or largest eigenvalues in magnitude), a subset of k approximated eigenvectors is chosen from among the $k + n_m$ ones to define a space that will be used to span U_k^{new}. Then, we describe in Theorem 1 the update of U_k^{new} and its image C_k^{new} with respect to A at restart of IB-BGCRO-DR.

THEOREM 1. *At restart of IB-BGCRO-DR, if we intend to deflate the space* $\mathrm{span}([U_k, \mathcal{V}_m]G_k)$*, where* $G_k = [g_1, \ldots, g_k]$ *is the set of vectors associated with the targeted eigenvalues, then the matrices* U_k^{new} *and* C_k^{new} *to be used for the next cycle are defined by*

$$U_k^{new} = \widehat{\mathcal{W}}_m G_k R^{-1} = [U_k, \mathcal{V}_m]G_k R^{-1}, \tag{2.21}$$

$$C_k^{new} = \widehat{\mathcal{V}}_{m+1} Q = \left[C_k, \mathcal{V}_m, [P_{m-1}, \widetilde{W}_m]\right] Q, \tag{2.22}$$

where Q *and* R *are the factors of the reduced* QR*-factorization of the tall and skinny matrix* $\underline{\mathcal{F}}_m G_k$*, which ensure that* $AU_k^{new} = C_k^{new}$ *and* $(C_k^{new})^H C_k^{new} = I_k$*.*

Proof. Let Q and R be the factors of the reduced QR-factorization of the tall and skinny matrix $\underline{\mathcal{F}}_m G_k$. Right multiplying G_k on both sides of (2.15) leads to $A\widehat{\mathcal{W}}_m G_k = \widehat{\mathcal{V}}_{m+1}\underline{\mathcal{F}}_m G_k = \widehat{\mathcal{V}}_{m+1} QR$ that is equivalent to $A\widehat{\mathcal{W}}_m G_k R^{-1} = \widehat{\mathcal{V}}_{m+1}\underline{\mathcal{F}}_m G_k R^{-1} = \widehat{\mathcal{V}}_{m+1} Q$, concluding the proof; as $\mathrm{span}(\widehat{\mathcal{W}}_m G_k R^{-1}) = \mathrm{span}(\widehat{\mathcal{W}}_m G_k)$, and $\widehat{\mathcal{V}}_{m+1} Q$ is the product of two matrices with orthonormal columns, so are its columns. ∎

COROLLARY 1. *The residual block at restart* $R_1^{new} = R_m^{old} = B - AX_1^{new}$ *with* $X_1^{new} = X_m^{old}$ *is orthogonal to* C_k^{new}*.*

Proof. $X_m^{old} = X_1 + \widehat{\mathcal{W}}_m Y_m$, where Y_m solves the least squares problem (2.18) so that $(\Lambda_m - \underline{\mathcal{F}}_m Y_m) \in (\mathrm{Range}(\underline{\mathcal{F}}_m))^{\perp} = \mathrm{Null}(\underline{\mathcal{F}}_m^H)$. We also have $R_m^{old} = \widehat{\mathcal{V}}_{m+1}(\Lambda_m - \underline{\mathcal{F}}_m Y_m)$, and consequently,

$$\begin{aligned}
(C_k^{new})^H R_m^{old} &= \left(\widehat{\mathcal{V}}_{m+1} Q\right)^H \left(\widehat{\mathcal{V}}_{m+1}(\Lambda_m - \underline{\mathcal{F}}_m Y_m)\right) \\
&= \left(\widehat{\mathcal{V}}_{m+1}\underline{\mathcal{F}}_m G_k R^{-1}\right)^H \left(\widehat{\mathcal{V}}_{m+1}(\Lambda_m - \underline{\mathcal{F}}_m Y_m)\right) \\
&= R^{-H} G_k^H \underbrace{\underline{\mathcal{F}}_m^H (\Lambda_m - \underline{\mathcal{F}}_m Y_m)}_{=0 \quad \text{because of (2.18)}} = 0.
\end{aligned}$$

2.5. A variant suited for flexible preconditioning. All the descriptions in the previous sections are naturally extended to the right preconditioning case with a fixed preconditioner M, and the central equality reads

$$A[U_k, M\mathcal{V}_m] = \left[C_k, \mathcal{V}_m, [P_{m-1}, \widetilde{W}_m]\right] \underline{\mathcal{F}}_m. \tag{2.23}$$

The least squares problem to be solved to compute the minimum norm solution becomes

$$Y_m = \underset{Y \in \mathbb{C}^{(k+n_m)\times p}}{\mathrm{argmin}} \left\| \Lambda_m - \underline{\mathcal{F}}_m Y \right\|_F,$$

and the solution is

$$X_m = X_1 + [U_k, M\mathcal{V}_m] Y_m.$$

If we denote by $\mathscr{M}_j$ a (possibly nonlinear) nonsingular preconditioning operator at iteration j and by $\mathscr{M}_j(\mathbb{V}_j)$ the action of $\mathscr{M}_j$ on a block vector $\mathbb{V}_j$, (2.23) translates into

$$A[U_k, \mathscr{Z}_m] = \left[C_k, \mathscr{V}_m, [P_{m-1}, \widetilde{W}_m]\right] \underline{\mathscr{F}}_m \text{ with } \mathscr{Z}_m = [\mathscr{M}_1(\mathbb{V}_1), \ldots, \mathscr{M}_m(\mathbb{V}_m)],$$

which can be written in a more compact form as

$$A\widehat{\mathscr{Z}_m} = \widehat{\mathscr{V}}_{m+1}\underline{\mathscr{F}}_m \text{ with } \widehat{\mathscr{Z}_m} = [U_k, \mathscr{Z}_m] \text{ and } \widehat{\mathscr{V}}_{m+1} = \left[C_k, \mathscr{V}_m, [P_{m-1}, \widetilde{W}_m]\right]. \tag{2.24}$$

The solution update is $X_m = X_1 + [U_k, \mathscr{Z}_m]Y_m$. For the sake of simplicity, we choose to keep the notation for quantities that have the same meaning as in the nonflexible case, but of course they will have different values.

In the context of flexible preconditioning many strategies for defining harmonic-Ritz vectors can be envisioned for GCRO-DR. Among those considered in [4], we follow the one with a lower computational cost required in solving the generalized eigenvalue problem, referred to as Strategy C in [4]. Furthermore, it also allows us to obtain companion properties in the flexible preconditioning case that are quite similar to the ones we have shown in the nonpreconditioned case in section 2.4. We refer the reader to [9, Appendix A] for two other strategies for approximating targeted eigen-information. Proposition 2 indicates that with an appropriate definition of the harmonic-Ritz vectors, all the properties of IB-BGCRO-DR extend to the flexible preconditioning variant denoted as IB-BFGCRO-DR.

PROPOSITION 2. *At the end of a cycle of the IB-BFGCRO-DR algorithm, if the deflation space is built on the harmonic-Ritz vectors* $\mathcal{W}_m g_i \in \operatorname{span}(\mathcal{W}_m)$ *of* $A\widehat{\mathscr{Z}_m}\mathcal{W}_m^\dagger$ *with respect to* $\mathcal{W}_m = [\mathcal{W}_k, \mathscr{V}_m] \in \mathbb{C}^{n\times(k+n_m)}$*, the following hold:*

1. *The harmonic-Ritz pairs* $(\theta_i, \mathcal{W}_m g_i)$ *for all restarts satisfy*

$$\underline{\mathscr{F}}_m^H \underline{\mathscr{F}}_m g_i = \theta_j \underline{\mathscr{F}}_m^H \widehat{\mathscr{V}}_{m+1}^H \mathcal{W}_m g_i \quad \text{for } 1 \le i \le k + n_m, \tag{2.25}$$

where

$$\widehat{\mathscr{V}}_{m+1}^H \mathcal{W}_m = \begin{bmatrix} C_k^H \mathcal{W}_k & 0_{k\times n_m} \\ \mathscr{V}_m^H \mathcal{W}_k & I_{n_m} \\ P_{m-1}^H \mathcal{W}_k & \\ \widetilde{W}_m^H \mathcal{W}_k & 0_{p\times n_m} \end{bmatrix} \in \mathbb{C}^{(k+n_m+p)\times(k+n_m)}.$$

2. *At restart, if* $G_k = [g_1, \ldots, g_k]$ *is associated with the* k *targeted eigenvalues, the matrices* $\mathcal{W}_k^{new}$, U_k^{new}, *and* C_k^{new} *to be used for the next cycle are updated by*

$$\mathcal{W}_k^{new} = \mathcal{W}_m G_k R^{-1} = [\mathcal{W}_k, \mathscr{V}_m] G_k R^{-1}, \tag{2.26}$$

$$\begin{aligned} U_k^{new} &= \widehat{\mathscr{Z}_m} G_k R^{-1} = [U_k, \mathscr{Z}_m] G_k R^{-1}, \\ C_k^{new} &= \widehat{\mathscr{V}}_{m+1} Q = \left[C_k, \mathscr{V}_m, [P_{m-1}, \widetilde{W}_m]\right] Q, \end{aligned} \tag{2.27}$$

   where Q *and* R *are the factors of the reduced QR-factorization of the tall and skinny matrix* $\underline{\mathscr{F}}_m G_k$*, ensuring* $AU_k^{new} = C_k^{new}$ *with* $(C_k^{new})^H C_k^{new} = I_k$.
3. *The residual at restart* $R_1^{new} = R_m^{old} = B - AX_1^{new}$ *with* $X_1^{new} = X_m^{old}$ *is orthogonal to* C_k^{new}.

Proof. The proof essentially follows the arguments developed for IB-BGCRO-DR described in section 2.4. We refer the reader to [9, Appendix B] for the details. ∎

We also mention that a closely related numerical technique that extends IB-BGMRES-DR in the flexible preconditioning context can be derived similarly. We refer the reader to [9, Appendix C], where the resulting new algorithm named IB-BFGMRES-DR is detailed and its properties are described.

3. Search space expansion policies governed by the stopping criterion. In this section we describe a few novel policies for expanding the search space that generalize the original one considered for inexact breakdown detection [19]. In particular, we first show how numerical criteria for detecting the partial convergence and expanding the search space can be tuned to ensure that a targeted threshold for a prescribed stopping criterion based on the individual backward error solution will be eventually satisfied. Second, we present how computational constraints can be taken into account and combined with any of the previous numerical criteria to best cope with the performance of the underlying computer architecture.

The partial convergence detection briefly described in section 2.3 ensures that if all the singular values of the least squares residual are smaller than the threshold τ, then all the linear system residual norms are also smaller than τ (i.e., p partial convergences have occurred). This is due to the inequality

$$\forall i \quad \|b^{(i)} - Ax_j^{(i)}\| \leq \|B - AX_j\| = \|\Lambda_j - \underline{\mathscr{F}}_j Y_j\| = \sigma_{\max}(\Lambda_j - \underline{\mathscr{F}}_j Y_j) < \tau, \tag{3.1}$$

which follows from the facts that the 2-norm of a matrix is an upper bound of the 2-norm of its individual columns and that $\widehat{\mathscr{V}}_{j+1}$ has orthonormal columns.

3.1. Search space expansion policy governed by η_b. A classical stopping criterion for the solution of a linear system $Ax = b$ is based on backward error analysis and consists of stopping the iteration when

$$\eta_b(x_j) = \frac{\|b - Ax_j\|}{\|b\|} \leq \varepsilon. \tag{3.2}$$

This criterion was considered in [1] where it was consequently proposed to define $\tau = \varepsilon \min_{i=1,\ldots,p} \|b^{(i)}\|$. With this choice, when the iteration complies with (3.1), we have

$$\eta_b(x_j^{(i)}) \leq \frac{\|b - Ax_j^{(i)}\|}{\min\limits_{i=1,\ldots,p} \|b^{(i)}\|} \leq \varepsilon. \tag{3.3}$$

When the different right-hand sides have very different norms in magnitude, the subspace expansion associated with this criterion might not be effective because the upper bound in (3.3) will not be tight. This leads to enlarging the search space with directions that are not relevant (generating useless computation). In that context a better choice would be to focus on the space expansion to reduce the residual associated with the right-hand side of the large norm. For that purpose, the idea is to perform the SVD not directly on the least squares residual but on its scaled least squares residual.

PROPOSITION 3. *Performing the SVD of the scaled least squares residuals* $(\Lambda_j - \underline{\mathscr{F}}_j Y_j) D_{b,\varepsilon}$ *with threshold* $\tau = 1$ *and* $D_{b,\varepsilon} = \varepsilon^{-1} diag(\|b^{(1)}\|^{-1}, \cdots, \|b^{(p)}\|^{-1})$ *ensures that when* p *partial convergences have occurred, so that the search space cannot be*

enlarged, each of the current individual iterates complies with the stopping criterion (3.2).

Proof. This is a direct consequence of the inequalities

$$\max_{i=1,\dots,p} \frac{\|b^{(i)} - Ax_j^{(i)}\|}{\varepsilon \|b^{(i)}\|} \leq \|(B - AX_j)D_{b,\varepsilon}\| = \|(\Lambda_j - \underline{\mathscr{F}}_j Y_j)D_{b,\varepsilon}\| \leq 1$$

and implies $\forall i\ \eta_b(x_j^{(i)}) \leq \varepsilon$. ∎

In some applications all the solutions associated with a block of right-hand sides do not need to be solved with the same accuracy. That is, we may have to solve a family of right-hand sides $B = [b^{(1)}, \dots, b^{(p)}]$ with individual convergence thresholds $\varepsilon^{(i)}$ for the solution associated with each right-hand side $b^{(i)}$ $(i = 1, \dots, p)$; thus we have the more general version of (3.2),

$$\eta_{b^{(i)}}(x_j^{(i)}) = \frac{\|b^{(i)} - Ax_j^{(i)}\|}{\|b^{(i)}\|} \leq \varepsilon^{(i)}. \tag{3.4}$$

In that context, the subspace expansion policy can be easily adapted to ensure the convergence for each individual accuracy.

COROLLARY 2. *Performing the SVD of the scaled least squares residuals* $(\Lambda_j - \underline{\mathscr{F}}_j Y_j)D_{b,\varepsilon_i}$ *with threshold* $\tau = 1$ *and* $D_{b,\varepsilon_i} = diag((\varepsilon_1\|b^{(1)}\|)^{-1}, \dots, (\varepsilon_p\|b^{(p)}\|)^{-1})$ *ensures that when p partial convergences have occurred each of the current individual iterates complies with the stopping criterion* (3.4).

3.2. Search space expansion policy governed by $\eta_{A,b}$. One can also adapt the expansion policy described in the previous section to the situation where the stopping criterion is based on the normwise backward error on A and b, defined by

$$\eta_{A,b}(x_j) = \frac{\|b - Ax_j\|}{\|b\| + \|A\|\ \|x_j\|} \leq \varepsilon. \tag{3.5}$$

It suffices to define accordingly the scaled least squares residuals in the SVD that is involved in the search space expansion. We notice that this type of stopping criterion will have a computational penalty as the iterates of all individual iterations have to be computed to calculate their norm.

COROLLARY 3. *Performing the SVD of the scaled least squares residual* $(\Lambda_j - \underline{\mathscr{F}}_j Y_j)D_{A,b,\varepsilon}$ *with threshold* $\tau = 1$ *and* $D_{A,b,\varepsilon} = \varepsilon^{-1} diag((\|A\|\|x_j^{(1)}\| + \|b^{(1)}\|)^{-1}, \dots, (\|A\|\|x_j^{(p)}\| + \|b^{(p)}\|)^{-1})$ *ensures that when p partial convergences have occurred, each of the current individual iterates complies with the stopping criterion* (3.5).

We do not further develop these ideas, but similarly we could define expansion policies where for each solution we can select either η_b or $\eta_{A,b}$ as a stopping criterion with an individual threshold setting.

The occurrence of p partial convergences is a sufficient condition that ensures the convergence of the p solution vectors, but the convergence might occur earlier, and a more classical stopping criterion can be accommodated at a low computational cost. Given that the norms of true residuals are very close to those of the least squares residuals when the loss of orthogonality of the generated block Krylov basis is not too serious, one can also check the convergence by looking at the norm of the least

squares residual, which is easy to compute. Let $Q_j^{LS} R_j^{LS}$ be a full QR-factorization of $\underline{\mathscr{F}}_j$ (i.e., Q_j^{LS} is unitary); then

$$\Lambda_j - \underline{\mathscr{F}}_j Y_j = Q_j^{LS} \begin{pmatrix} 0_{(n_j+k)\times p} \\ R_j^{\ell s} \end{pmatrix}, \tag{3.6}$$

where $R_j^{\ell s} \in \mathbb{C}^{p\times p}$ are the last p rows of $(Q_j^{LS})^H \Lambda_j$ so that $\|b^{(i)} - Ax_j^{(i)}\| = \|R_j^{\ell s}(:,i)\|$. Those residual norm calculations are part of the stopping criterion based on η_b or $\eta_{A,b}$.

3.3. Search space expansion policy governed by computational performance. Based on any of these expansion policies, the discarded directions at a given iteration might be reintroduced in a subsequent one; thereby we can trade on the considered numerical policy and select for the subspace expansion only a subset of those eligible. In particular, it might be relevant to choose a prescribed block size p^{CB} (here the superscript ^{CB} stands for computational blocking) that is best suited to cope with the computational features on a given platform rather than selecting the numerical block size p_{j+1} defined as the number of singular values greater than or equal to the prescribed threshold $\tau = 1$. In that respect, we consider a subspace expansion policy so that the block size at the end of step j is defined as $p_{j+1}^{CB} = \min(p^{CB}, p_{j+1})$. We refer to this variant as inexact breakdown block GCRO-DR with computational blocking (denoted by IB-BGCRO-DR-CB).

Note that all the subspace expansion policies discussed in section 3 could be applied to any other block minimum residual norm methods equipped with the partial convergence detection, such as the IB-BGMRES [19] and IB-BGMRES-DR [1] algorithms.

4. Remarks on some computational and algorithmic aspects. The mathematical description in the previous section assumes exact calculation. In practice, the numerical behavior of the algorithms depends on the numerical algorithms selected to perform the computation in finite precision arithmetic. In particular, all the above descriptions assume the orthonormality of the residual basis; the orthonormality ensures the norm equality of the true linear system residual and their least squares counterpart, which governs the numerical search space expansion policies described in the previous section. In our implementation, for the block Arnoldi procedure (See Algorithm 1), we consider the block modified Gram–Schmidt (BMGS) algorithm with reduced QR-factorization based on Householder reflections of the final tall and skinny block (referred to as (BMGS $\circ$ HouseQR) in [3]). In addition, at restart the reorthogonalization of the recycling space C_k and of the initial block residual vector $[\mathbb{V}_1, P_0]$ in (4.2) is performed a vector at a time using modified Gram–Schmidt. For the sake of conciseness, we do not give full technical details of what we briefly present in the core of the paper, but we sometimes refer to a particular part in the appendices.

4.1. Inexact breakdown and re-orthogonalization at restart. For the sake of simplicity, in the previous sections we made the assumption that the initial residual block was of full rank. In practice, this constraint can be removed by applying the partial convergence detection to the initial residual block. In that case, only a subspace of the space spanned by the columns of the initial residual block will be selected to define the first search space, and the discarded directions are kept in the basis of the residual space. This has the following two main consequences:

1. The first iteration needs some extra attention to set up the initial basis $\mathbb{V}_1$ and discarded directions P_0 defined in (2.9).

2. A consequence of having discarded directions in the first search space is that the projection of the initial residual block in the residual space that defines the right-hand side of the least squares residual solved at each block iteration will no longer have the nested block structure that is expanded by a $p \times p$ zero block at each block iteration as presented in (2.18).

Without loss of generality, let us present the partial convergence detection and re-orthogonalization at restart where the recycling subspaces U_k^{new} and C_k^{new} are defined by (2.21) and (2.22), respectively, so that mathematically $AU_k^{new} = C_k^{new}$ and $(C_k^{new})^H C_k^{new} = I_k$, and the initial residual block $R_1^{new} = R_1$ in Corollary 1 is orthogonal to C_k^{new}. For a prescribed stopping criterion and convergence threshold, let us denote by D_ε the diagonal matrix used to select the space expansion described in section 3. Let

$$R_1 D_\varepsilon = [\mathbb{V}_1^{new}, P_0^{new}] \begin{bmatrix} \Sigma_{p_1} & \\ & \Sigma_{q_1} \end{bmatrix} \mathbb{V}_{R_1}^H = [\mathbb{V}_1^{new}, P_0^{new}] \hat{\Lambda}_1^{'}, \tag{4.1}$$

where $\mathbb{V}_1^{new} \in \mathbb{C}^{n \times p_1}$, $P_0^{new} \in \mathbb{C}^{n \times q_1}$ with $p_1 + q_1 = p$, and Σ_{p_1} contains the p_1 singular values of $R_1 D_\varepsilon$ greater than or equal to the prescribed τ, and Σ_{q_1} contains the ones smaller than τ. We first perform an MGS re-orthogonalization of the columns of $[C_k^{new}, [\mathbb{V}_1^{new}, P_0^{new}]]$ that reads

$$[C_k^{new}, [\mathbb{V}_1^{new}, P_0^{new}]] = [C_k, [\mathbb{V}_1, P_0]] \begin{bmatrix} R_{11} & R_{12} \\ & R_{22} \end{bmatrix}, \tag{4.2}$$

where all the columns of $[C_k, [\mathbb{V}_1, P_0]]$ are orthogonal to one another, and $\left[\begin{smallmatrix} R_{11} & R_{12} \\ & R_{22} \end{smallmatrix}\right] \in \mathbb{C}^{(k+p)\times(k+p)}$ is an upper triangular matrix with $R_{11} \in \mathbb{C}^{k\times k}$ and $R_{22} \in \mathbb{C}^{p\times p}$. Next, we update $U_k = U_k^{new} R_{11}^{-1}$ to satisfy (2.1), and $\mathscr{V}_1 = \mathbb{V}_1$ will serve to span the first search space and P_0 will be abandoned for this first block iteration that will be run as follows:

1. Form $W_1 = A\mathbb{V}_1$ and orthogonalize it (using BMGS $\circ$ HouseQR) against the set of orthonormal vectors that are part of the residual space $[C_k, \mathbb{V}_1, P_0]$ which enables the computation of the entries of $\mathcal{B}_1 = C_k^H W_1$, $\mathscr{L}_{1,1} = \mathbb{V}_1^H W_1$, and $E_1 = P_0^H W_1$.
2. The resulting block $\bar{W}_1$ formally reads $\bar{W}_1 = W_1 - C_k\mathcal{B}_1 - \mathbb{V}_1\mathscr{L}_{1,1} - P_0 E_1$, with $\bar{W}_1 = \widetilde{W}_1 D_1$ being its reduced QR-factorization.
3. In matrix form the above relations also read

$$W_1 = A\mathbb{V}_1 = \left[C_k, \mathbb{V}_1, [P_0, \widetilde{W}_1]\right] \begin{bmatrix} \mathcal{B}_1 \\ \mathscr{L}_{1,1} \\ E_1 \\ D_1 \end{bmatrix},$$

so that we have the first Arnoldi-like relation

$$A[U_k, \mathbb{V}_1] = \left[C_k, \mathbb{V}_1, [P_0, \widetilde{W}_1]\right] \underline{\mathscr{F}}_1, \tag{4.3}$$

with

$$\underline{\mathscr{F}}_1 = \begin{bmatrix} I_k & \mathcal{B}_1 \\ 0_{(p_1+p)\times k} & \begin{matrix} \mathscr{L}_{1,1} \\ \widetilde{\mathbb{H}}_1 \end{matrix} \end{bmatrix} \in \mathbb{C}^{(k+p_1+p)\times(k+p_1)} \text{ and } \widetilde{\mathbb{H}}_1 = \begin{bmatrix} E_1 \\ D_1 \end{bmatrix} \in \mathbb{C}^{p\times p_1}.$$

4. Next, define the minimum norm solution $X_2 = X_1 + [U_k, \mathbb{V}_1]Y$, and note that R_1 belongs to the space $[C_k, \mathbb{V}_1, P_0, \widetilde{W}_1]$ where its components in this orthogonal basis are given by $[C_k, \mathbb{V}_1, P_0, \widetilde{W}_1]^H R_1$. From (4.3) we have

$$\begin{aligned}\|B - AX_2\|_F &= \|R_1 - A\,[U_k, \mathbb{V}_1]\,Y\|_F = \|R_1 - [C_k, \mathbb{V}_1, P_0, \widetilde{W}_1]\underline{\mathscr{F}}_1 Y\|_F \\ &= \|[C_k, \mathbb{V}_1, P_0, \widetilde{W}_1]^H R_1 - \underline{\mathscr{F}}_1 Y\|_F \\ &= \|[C_k, \mathbb{V}_1, P_0, \widetilde{W}_1]^H [\mathbb{V}_1^{new}, P_0^{new}]\hat{\Lambda}_1 - \underline{\mathscr{F}}_1 Y\|_F,\end{aligned}$$

and then from (4.1), we have

$$R_1 = [\mathbb{V}_1^{new}, P_0^{new}]\hat{\Lambda}_1' D_\varepsilon^{-1} = [\mathbb{V}_1^{new}, P_0^{new}]\hat{\Lambda}_1 \text{ with } \hat{\Lambda}_1 = \hat{\Lambda}_1' D_\varepsilon^{-1}, \tag{4.4}$$

so that from (4.2), the right-hand side of the above least squares residual reads

(4.5)

$$\begin{aligned}\Lambda_1 &= [C_k, \mathbb{V}_1, P_0, \widetilde{W}_1]^H [\mathbb{V}_1^{new}, P_0^{new}]\hat{\Lambda}_1 \\ &= [C_k, \mathbb{V}_1, P_0, \widetilde{W}_1]^H [C_k R_{12} + [\mathbb{V}_1, P_0] R_{22}]\hat{\Lambda}_1 \\ &= \left([C_k, \mathbb{V}_1, P_0, \widetilde{W}_1]^H C_k R_{12} + [C_k, \mathbb{V}_1, P_0, \widetilde{W}_1]^H [\mathbb{V}_1, P_0] R_{22}\right)\hat{\Lambda}_1 \\ &= \begin{bmatrix} R_{12} \\ 0_{(p_1+p)\times p} \end{bmatrix}\hat{\Lambda}_1 + \begin{bmatrix} 0_{k\times p_1} & 0_{k\times q_1} \\ I_{p_1} & 0_{p_1\times q_1} \\ 0_{q_1\times p_1} & I_{q_1} \\ 0_{p_1\times p_1} & 0_{p_1\times q_1} \end{bmatrix} R_{22}\hat{\Lambda}_1 \in \mathbb{C}^{(k+p_1+p)\times p}.\end{aligned}$$

5. Compute Y_1, the solution of the first new least squares problem:

$$Y_1 = \underset{Y\in\mathbb{C}^{(k+p_1)\times p}}{\operatorname{argmin}} \|\Lambda_1 - \underline{\mathscr{F}}_1 Y\|_F.$$

6. Execute the search space expansion policy following the IB principles:
   (a) Compute the SVD of the scaled least squares residual

$$\begin{aligned}(\Lambda_1 - \underline{\mathscr{F}}_1 Y_1)\, D_\varepsilon = \mathbb{U}_{1,L}\Sigma_1 \mathbb{V}_{1,R}^H + \mathbb{U}_{2,L}\Sigma_2 \mathbb{V}_{2,R}^H, \\ \text{where } \sigma_{\min}(\Sigma_1) \geq 1 > \sigma_{\max}(\Sigma_2).\end{aligned}$$

   (b) Compute $\mathbb{W}_1$ and $\mathbb{W}_2$ such that $\operatorname{Range}(\mathbb{W}_1) = \operatorname{Range}(\mathbb{U}_1^{(2)}) \in \mathbb{C}^{p\times p_2}$ with $\mathbb{U}_{1,L} = \begin{pmatrix} \mathbb{U}_1^{(1)} \\ \mathbb{U}_1^{(2)} \end{pmatrix} \in \mathbb{C}^{(k+p_1+p)\times p_2}$, $[\mathbb{W}_1, \quad \mathbb{W}_2]$ is unitary, and $\mathbb{W}_2 \in \mathbb{C}^{p\times q_2}$ with $p_2 + q_2 = p$.

   (c) Compute the new orthonormal matrices $\mathbb{V}_2$ and P_1 as

$$\mathbb{V}_2 = [P_0, \widetilde{W}_1]\mathbb{W}_1 \in \mathbb{C}^{n\times p_2},\ P_1 = [P_0, \widetilde{W}_1]\mathbb{W}_2 \in \mathbb{C}^{n\times q_2},$$

   and compute as well as the last block row matrix $\mathscr{L}_{2,:}$ of $\underline{\mathscr{L}}_1$ and $\mathbb{G}_1$ as

$$\mathscr{L}_{2,:} = \mathbb{W}_1^H \widetilde{\mathbb{H}}_1 \in \mathbb{C}^{p_2\times p_1},\ \mathbb{G}_1 = \mathbb{W}_2^H \widetilde{\mathbb{H}}_1 \in \mathbb{C}^{q_2\times p_1}.$$

7. Set $\underline{\mathscr{L}}_1 = \begin{pmatrix} \mathscr{L}_1 \\ \mathscr{L}_{2,:} \end{pmatrix} \in \mathbb{C}^{(p_1+p_2)\times p_1} = \mathbb{C}^{n_2\times p_1}$.

Whenever a partial convergence is detected in R_1, some of its components (along P_0^{new}) are first discarded but could be reintroduced in some subsequent iterations. One of the consequences of this is that the last q_1 columns of the least squares right-hand side problem evolve from one iteration to the next, depending on how some of the P_0^{new} directions are reintroduced in the search space along the iterations. There is a way to incrementally update the least squares right-hand side, and this is discussed in the next proposition.

PROPOSITION 4. *At each iteration of IB-BGCRO-DR, the new least squares problem reads*

(4.6)
$$Y_{j+1} = \underset{Y \in \mathbb{C}^{(k+n_{j+1})\times p}}{\operatorname{argmin}} \left\| \Lambda_{j+1} - \underline{\mathscr{F}}_{j+1} Y \right\|_F, \quad \Lambda_{j+1} \in \mathbb{C}^{(k+n_{j+1}+p)\times p}, \quad j = 0, 1, 2, \ldots,$$

with the updated right-hand sides being

(4.7)
$$\Lambda_{j+1} = \begin{bmatrix} R_{12} \\ 0_{(n_j+p+p_{j+1})\times p} \end{bmatrix} \hat{\Lambda}_1 + \begin{bmatrix} 0_{k\times p_1} & 0_{k\times q_1} \\ \begin{bmatrix} I_{p_1} \\ 0_{(n_j+p-p_1)\times p_1} \end{bmatrix} & \Phi_{j+1} \\ 0_{p_{j+1}\times p_1} & 0_{p_{j+1}\times q_1} \end{bmatrix} R_{22}\ \hat{\Lambda}_1,$$

where $\Phi_{j+1} = \begin{bmatrix} \Phi_j(1{:}n_j,:) \\ [\mathbb{W}_1,\mathbb{W}_2]^H \begin{bmatrix} \Phi_j(n_j+1{:}n_j+q_j,:) \\ 0_{p_j\times q_1} \end{bmatrix} \end{bmatrix} \in \mathbb{C}^{(n_j+p)\times q_1}$ *for* $j = 0, 1, 2, \ldots,$ *with* $\Phi_1 = \begin{bmatrix} 0_{p_1\times q_1} \\ I_{q_1} \end{bmatrix} \in \mathbb{C}^{p\times q_1}$ *and* $q_j = p - p_j (j > 0)$*;* $[\mathbb{W}_1, \mathbb{W}_2]$ *is unitary as defined in the search space expansion algorithm based on IB principles; and* $R_{12} \in \mathbb{C}^{k\times p}$ *and* $R_{22} \in \mathbb{C}^{p\times p}$ *are two block components of the upper triangular matrix as shown in the right-hand side of* (4.2)*.*

Proof. We refer the reader to Appendix A for details of the proof. ∎

Based on the above discussions, the IB-BGCRO-DR algorithm with partial convergence detection in the initial residual block and updated right-hand sides of the least squares residual is presented as Algorithm 2 for solving a series of linear systems with slowly changing left-hand sides.

4.2. Solution of the least squares problem and cheap SVD calculation of the scaled least squares residual. Computing the full QR-factorization of the matrices involved in the least squares problems allows us to reuse its Q factor to compute the SVD of the least squares residual using a QR-SVD algorithm such that the actual SVD decomposition is performed on a $p\times p$ block $R_j^{\ell s} D_\varepsilon$, where $R_j^{\ell s}$ appears as in the right-hand side of (3.6), at each iteration (we refer the reader to Appendix B for the details of this calculation). Note that this observation applies naturally to the IB-BGMRES [19] and IB-BGMRES-DR [1] algorithms as well.

5. Numerical experiments. In the following sections we illustrate different numerical features of the novel algorithm introduced above. For the sake of comparison, in some of the experiments we also display results of closely related block methods such as BGCRO-DR [17, 21, 28] or IB-BGMRES-DR [1]. All the numerical experiments have been run using a MATLAB prototype, so that the respective performances of the algorithms are evaluated in terms of the number of matrix-vector products, denoted as *mvps* (and preconditioner applications in the preconditioned case) required to converge.

Algorithm 2. IB-BGCRO-DR for slowly changing left-hand sides and massive number of right-hand sides.

Require: $A \in \mathbb{C}^{n\times n}$ left-hand side of current family (assumed to not vary much compared to previous one)
Require: $B \in \mathbb{C}^{n\times p}$ the block of right-hand-sides and $X_0 \in \mathbb{C}^{n\times p}$ the block initial guess
Require: m maximum number of Arnoldi steps within a cycle
Require: p^{CB} a given constant number satisfying $1 \leq p^{CB} \leq p$ for computational blocking
Require: $D_\varepsilon \in \mathbb{C}^{p\times p}$ a diagonal matrix used to select the space expansion described in section 3
Require: $U_k, C_k \in \mathbb{C}^{n\times k}$ the recycling subspaces assumed to be empty for the first family and obtained after solving previous slow-changing family

Compute $R_0 = B - AX_0$
/ Some families have already been solved ? */*
if the recycling space is not empty, $U_k \neq 0$ **then**
  Apply the reduced QR-factorization to AU_k for updating U_k and C_k for the current family such that the U_k and C_k satisfy (2.1) and (2.2). Compute R_1 and X_1 as described in (2.4)
else
  Set $R_1 = R_0$, $X_1 = X_0$, $U_k = 0$, $C_k = 0$
end if
/ Loop over the restarts */*
while the stopping criterion based on section 3.1 or 3.2 is not met **do**
  Apply partial convergence detection in the scaled (least squares) residual block following section 4.1
  / Arnoldi loop */*
  for $j = 2, 3, \ldots, m$ **do**
    Orthogonalize $A\mathbb{V}_j$ against C_k as $W_j = (I - C_kC_k^H)A\mathbb{V}_j$. Then orthogonalize W_j against previous block orthonormal vector $\mathscr{V}_j = [\mathbb{V}_1, \ldots, \mathbb{V}_j]$ as

$$W_j = A\mathbb{V}_j - C_kC_k^HA\mathbb{V}_j - \mathscr{V}_j\mathscr{L}_{1,1:j}, \text{ where } \mathscr{L}_{1,1:j} = \mathscr{V}_j^H(W_j) = \mathscr{V}_j^H(A\mathbb{V}_j) \text{ is a block column matrix}$$

    Set $\mathscr{L}_j = \begin{bmatrix}\underline{\mathscr{L}}_{j-1} & \mathscr{L}_{1,1:j}\end{bmatrix} \in \mathbb{C}^{n_j\times n_j}$, $\mathcal{B}_j = \begin{bmatrix}\mathcal{B}_{j-1} & C_k^HA\mathbb{V}_j\end{bmatrix} \in \mathbb{C}^{k\times n_j}$
    Orthogonalize W_j against P_{j-1} and carry out its reduced QR-factorization as

$$\widetilde{W}_jD_j = W_j - P_{j-1}E_j, \text{ where } E_j = P_{j-1}^HW_j$$

    Compute Y_j by solving the least squares problem described in (2.11) (or (4.6)) with $\underline{\mathscr{F}}_j$ shown in (2.12) composed of $\mathscr{F}_j$ and $\mathbb{H}_j$ but with the updated right-hand side Λ_j as shown in (4.7) instead
    if the stopping criterion is met **then**
      return $X_j = X_1 + [U_k, \mathscr{V}_j]Y_j$, U_k, and C_k
    end if
    Singular value decomposition of the residuals scaled by D_ε

$$(\Lambda_j - \underline{\mathscr{F}}_jY)D_\varepsilon = \mathbb{U}_{1,L}\Sigma_1\mathbb{V}_{1,R}^H + \mathbb{U}_{2,l}\Sigma_2\mathbb{V}_{2,R}^H \text{ with } \sigma_{\min}(\Sigma_1) \geq 1 > \sigma_{\max}(\Sigma_2)$$

    if Computational blocking of section 3.3 is activated **then**
      $\mathbb{U}_{1,L} = \mathbb{U}_{1,L}(:, 1 : p_j^{CB})$ with $p_j^{CB} = min(p^{CB}, nl_{\Sigma_1})$, nl_{Σ_1} refers to column number of Σ_1
    end if
    Following item 6 described in section 4.1 for computing $\mathbb{W}_1$ and $\mathbb{W}_2$
    Compute orthonormal matrices $\mathbb{V}_{j+1}$ and P_j, the last block row matrix $\mathscr{L}_{j+1,:}$ of $\underline{\mathscr{L}}_j$, and G_j as

$$\mathbb{V}_{j+1} = \begin{bmatrix}P_{j-1}, \widetilde{W}_j\end{bmatrix}\mathbb{W}_1, P_j = \begin{bmatrix}P_{j-1}, \widetilde{W}_j\end{bmatrix}\mathbb{W}_2, \mathscr{L}_{j+1,:} = \mathbb{W}_1^H\mathbb{H}_j, \mathbb{G}_j = \mathbb{W}_2^H\mathbb{H}_j, \underline{\mathscr{L}}_j = \begin{pmatrix}\mathscr{L}_j \\ \mathscr{L}_{j+1,:}\end{pmatrix}$$

  end for
  / Restart procedure */*
  Compute the solution X_m as described in (2.16) and residual R_m according to (2.17)
  Compute the targeted harmonic-Ritz vectors $G_k = [g_1, \ldots, g_k]$ by solving the generalized eigenvalue problem (2.19) described in Proposition 1
  Update the values of U_k and C_k, respectively, by (2.21) and (2.22) described in Theorem 1
  Restart with $X_1 = X_m$, $\widehat{\mathscr{V}}_{m+1}$, $R_1^{LS} = \Lambda_m - \underline{\mathscr{F}}_mY_m$ ($R_1 = R_m = \widehat{\mathscr{V}}_{m+1}R_1^{LS}$)
end while
return X_j for approximation of the current family; U_k, C_k for the next family to be solved

For each set of blocks of right-hand sides, referred to as a family, the block initial guess is equal to $0 \in \mathbb{C}^{n\times p}$, where p is the number of right-hand sides. The block right-hand side $B = [b^{(1)}, b^{(2)}, \ldots, b^{(p)}] \in \mathbb{C}^{n\times p}$ is composed of p linearly independent vectors generated randomly (using the same random seed when block methods are compared). While any part of the spectrum could be considered to define the recycling space, we consider for all the experiments the approximated eigenvectors associated with the k smallest approximated eigenvalues in magnitude. The maximum dimension of the search space in each cycle is set to be $m_d = 15 \times p$. To illustrate the potential benefit of IB-BGCRO-DR when compared to another block solver, we consider the overall potential gain when solving a sequence of ℓ families defined as

$$\text{Gain } (\ell) = \frac{\sum_{s=1}^{\ell} \#mvps \text{ (method)}^{(s)}}{\sum_{s=1}^{\ell} \#mvps \text{ (IB-BGCRO-DR)}^{(s)}}. \tag{5.1}$$

5.1. Benefits of recycling between the families. To illustrate the benefits of recycling spectral information from one family to the next as well as the computational savings due to the partial convergence detection mechanism, we first report on experiments with BGCRO-DR, IB-BGCRO-DR, and IB-BGMRES-DR on a series of linear systems with a constant left-hand side.

Following in the spirit of the test examples considered in [12], we consider a bidiagonal matrix of size 5000 with upper diagonal unity so that its spectrum is defined by the diagonal entries $0.1, 1, 2, 3, \ldots, 4999$; this is denoted as Matrix 1. We consider experiments with a family size $p = 20$ and a recycled space size $k = 30$, and where the maximal dimension of the search space is $m_d = 300$.

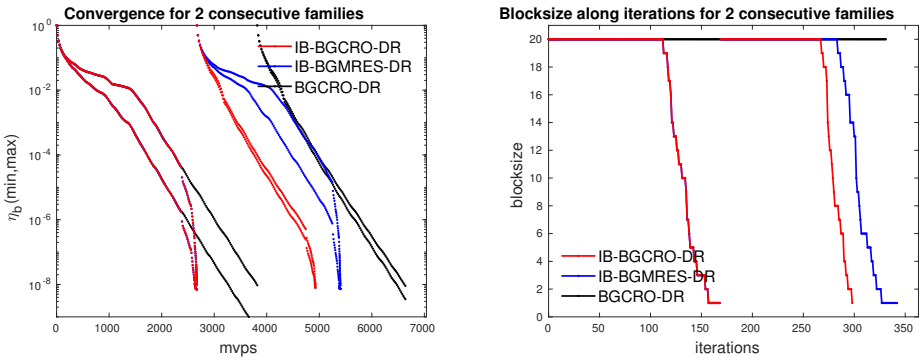

FIG. 1. *Comparison history for section* 5.1. *IB-BGCRO-DR with BGCRO-DR and IB-BGMRES-DR by solving Matrix* 1 *($p = 20$, $m_d = 300$, and $k = 30$). Left: convergence histories of the largest/smallest backward errors $\eta_{b^{(i)}}$ at each mvps for* 2 *consecutive families. Right: varying blocksize (i.e., p_j) along the iterations.*

In the left plot of Figure 1 we display the convergence histories for solving two consecutive families with the η_b-based stopping criterion. Several observations can be made. Because IB-BGMRES-DR, IB-BGCRO-DR, and BGCRO-DR do not have a deflation space to start with for the first family, the convergence histories of these three solvers overlap as long as no partial convergence is detected. After this first partial convergence, the convergence rate of IB-BGCRO-DR and IB-BGMRES-DR becomes faster (in terms of *mvps*) than that of BGCRO-DR, and the former two convergence histories mostly overlap as the two IB solvers remain mathematically equivalent. For the second and subsequent families, the capability of starting with a

deflation space shows its benefit for BGCRO-DR and IB-BGCRO-DR. It is because IB-BGMRES-DR needs a few restarts to capture this spectral information again and to refine it in its subsequent search spaces construction process; eventually it exhibits a convergence rate similar to the BGCRO-DR counterpart. For the sake of comparison and to illustrate the benefit of the partial convergence detection we also display the convergence histories of BGCRO-DR, which always requires more *mvps* compared to its IB counterpart. Those extra *mvps* mostly enable us to improve the solution quality for some right-hand sides beyond the targeted accuracy.

To visualize the effect of the partial convergence detection, we report in the right plot of Figure 1 the size of search space expansion p_j as a function of the iterations. Because BGCRO-DR does not implement the partial convergence detection, its search space is increased by $p = 20$ at each iteration. For the other two block IB-solvers, the block size monotonically decreases to 1. Note that the partial convergence detection is implemented in the initial (least squares) residual block in IB-BGCRO-DR, and thus its block size does not jump back to the original block size p at restart. By construction, IB-BGMRES-DR implements the partial convergence detection at restart so that the same observation applies.

TABLE 1

Numerical results in terms of both mvps and its for section 5.1 *with Matrix* 1 *($p = 20$, $m_d = 300$, and $k = 30$).*

Number of families	Method	*mvps*	*its*
2	BGCRO-DR	6640	332
	IB-BGMRES-DR	5404	343
	IB-BGCRO-DR	**4928**	**299**
20	BGCRO-DR	56940	2847
	IB-BGMRES-DR	53772	3454
	IB-BGCRO-DR	**45652**	**2637**

A summary of the *mvps* and the number of block iterations (referred to as *its*) is given in Table 1 that shows the benefit of using IB-BGCRO-DR.

In the rest of this paper, Matrix 1 is chosen as the constant left-hand side in sections 5.2–5.4, in which the related parameters are likewise set to be $p = 20$, $k = 30$, and $m_d = 300$ by default.

5.2. Subspace expansion governed by the convergence criterion $\eta_{A,b}$. In this section we show the capability of the novel subspace expansion policy to drive the individual backward errors $\eta_{A,b}$ down to different accuracies and its benefit with respect to the original BGCRO-DR method. In Figure 2, we display the convergence histories of the IB and IB-free methods for three different convergence thresholds, from the less stringent on the left to the most stringent on the right. We can first observe that the first iteration, where the partial convergence detection starts to act, depends on the targeted accuracy as can have been expected from the associated threshold on the singular values of the least squares residual. The second interesting observation is that IB-BGCRO-DR is able to decrease $\eta_{A,b}$ to a very low value close to the machine epsilon, that is, $\mathcal{O}(10^{-16})$. This latter result mostly reveals the orthogonality quality of the residual space basis computed by (BMGS $\circ$ HouseQR) in the block Arnoldi implementation and the re-orthogonalization using MGS between all the columns of the recycling subspace C_k and the initial block Arnoldi basis at restart. This ensures that the least squares residual norms are quite close to the linear system residual ones. This latter fact ensures the relevance of the space expansion policy that monitors the

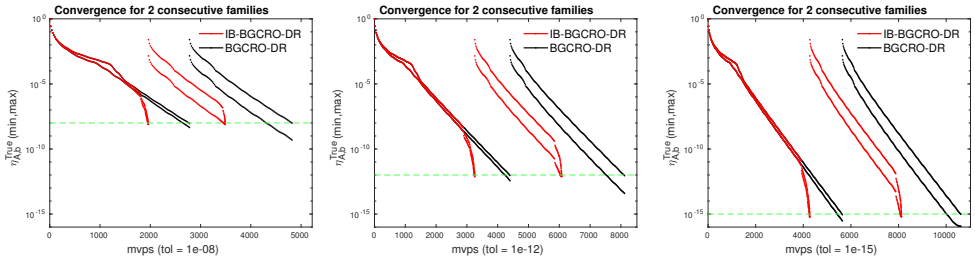

FIG. 2. *Convergence histories of the largest/smallest* $\eta_{A,b^{(i)}}(x_j^{(i)})$ *at each mvps for* 2 *consecutive families for section* 5.2 *with different convergence thresholds. Comparison of IB-BGCRO-DR with BGCRO-DR by solving Matrix* 1 *(*$p = 20$, $m_d = 300$, *and* $k = 30$*).*

linear system residual norms through the least squares residual ones. To illustrate the orthonormal quality of the basis $\widehat{\mathscr{V}}_{j+1} = \left[C_k, \mathscr{V}_j, [P_{j-1}, \widetilde{W}_j]\right]$, we display in Figure 3 the loss of orthogonality along *mvps* that is defined by

$$\text{Loss-Orth} = \left\| \widehat{\mathscr{V}}_{j+1}^H \widehat{\mathscr{V}}_{j+1} - I_{j+1} \right\|. \tag{5.2}$$

In a quite similar manner to MGS-GMRES that is backward-stable [14], it can be observed that the loss of orthogonality mostly appears when the solutions of the linear systems converge. Note that without the re-orthogonalization at restart, the loss of orthogonality tends to accumulate along restart, which prevents the value of Loss-Orth from being close to the machine epsilon. We refer the reader to [9, Figure 5.7] for the corresponding results without applying re-orthogonalization to all the columns of $[C_k, [\mathbb{V}_1, P_0]]$ at restart.

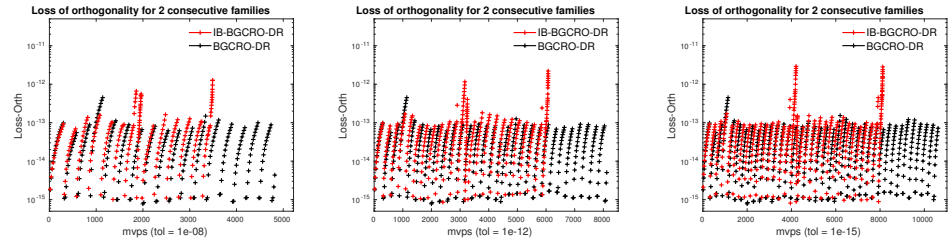

FIG. 3. *Loss-Orth defined in* (5.2) *of GCRO-variants with stopping criterion based on* $\eta_{A,b^{(i)}}(x_j^{(i)})$ *at each mvps for* 2 *consecutive families for section* 5.2 *with different convergence thresholds. Comparison of IB-BGCRO-DR with BGCRO-DR for solving Matrix* 1 *(*$p = 20$, $m_d = 300$, *and* $k = 30$*).*

5.3. Subspace expansion policy for individual convergence thresholds for η_b. To illustrate this feature, we consider a family of p right-hand sides and a convergence threshold 10^{-4} for the first $p/2$ right-hand sides and 10^{-8} for the last $p/2$ ones. To estimate the computational benefit of this feature, we also compare with calculations where all the right-hand sides are solved with the most stringent threshold, that is, 10^{-8}. In the left part of Figure 4, we display the convergence histories for 3 successive families. The variant that controls the individual threshold is denoted as IB-BGCRO-DR-VA, where VA stands for variable accuracy. It can be seen

that the numerical feature works well and that the envelope of the backward errors has the expected shape, that is, the minimum backward error decreases to 10^{-8}, while the maximum one (associated with the first $p/2$ solutions) only decreases to 10^{-4}. If we compare the convergence histories of IB-BGCRO-DR and IB-BGCRO-DR-VA, it can be seen that the slope of IB-BGCRO-DR-VA is deeper than that of IB-BGCRO-DR once the first $p/2$ solutions have converged; after this point IB-BGCRO-DR-VA somehow focuses on the new directions (produced by *mvps* given for the x-axis) to reduce the residual norms of the remaining $p/2$ solutions that have not yet converged. The right plot of Figure 4 shows the computational gain induced by the individual control of the accuracy compared to the situation where all the right-hand sides would have been solved to the most stringent stopping criterion threshold if this feature were not designed. In this case the individual monitoring of the convergence saves around 45% of *mvps* in this example. Those results are summarized in Table 2.

We refer the reader to [9, Figure F.1 and Table F.1 of Appendix F] for an illustration of extending such individual control to the block solver IB-BGMRES-DR that can also accommodate this feature.

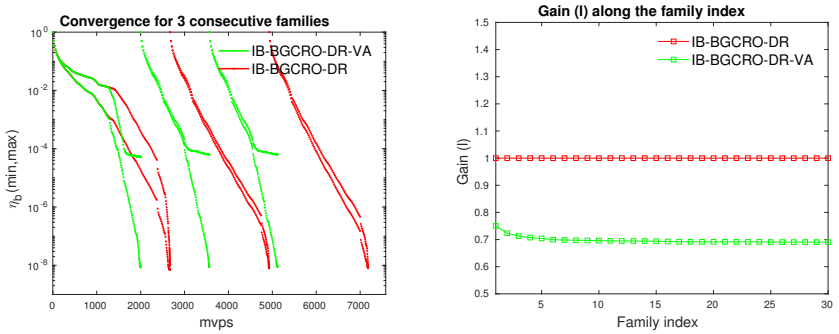

FIG. 4. *Comparison of IB-BGCRO-DR to IB-BGCRO-DR-VA for section* 5.3 *with Matrix* 1 *($p = 20$, $m_d = 300$, and $k = 30$). Left: convergence histories of the largest/smallest backward errors $\eta_{b^{(i)}}$ at each mvps for* 3 *consecutive families. Right: gain (ℓ) defined in* (5.1) *of IB-BGCRO-DR-VA to IB-BGCRO-DR versus family index.*

TABLE 2

Numerical results of IB-BGCRO-DR with fixed/varying accuracy for each right-hand side in terms of mvps and its for section 5.3*, where the coefficient matrix is Matrix* 1 *with $p = 20$, $m_d = 300$, and $k = 30$.*

Number of families	Method	*mvps*	*its*
3	IB-BGCRO-DR	7182	428
	IB-BGCRO-DR-VA	**5119**	**395**
30	IB-BGCRO-DR	68263	3932
	IB-BGCRO-DR-VA	**47143**	**3566**

5.4. Expansion policy governed by computational performance. As discussed in section 3.3, only a subset of the candidate directions exhibited by the partial convergence detection mechanism can be eventually selected to expand the search space at the next block iteration; we denote this maximum size as p^{CB} and refer to this variant as IB-BGCRO-DR-CB, where CB stands for computational blocking. In Table 3 we show the effect of this algorithmic parameter on *mvps* and *its* for the

solutions of 3 and 30 families with Matrix 1 when p^{CB} varies from 1 to 15 for a number of right-hand sides $p = 20$. Generally, the smaller p^{CB} is, the smaller *mvps*, but the larger *its*. Although reported only on one example, this trend has been observed in all our numerical experiments. Depending on the computational efficiency or cost of the *mvps* with respect to the computational weight of the least squares problem and SVD of the scaled least squares residual, this gives opportunities to monitor the overall computational effort needed to complete the solution.

TABLE 3
Numerical results of IB-BGCRO-DR and IB-BGCRO-DR-CB for $p^{CB} = 1, 5, 10, 15$ in terms of mvps and its for section 5.4, where the coefficient matrix is Matrix 1 with $p = 20$, $m_d = 300$, and $k = 30$.

Number of families	Method	*mvps*	*its*
3	IB-BGCRO-DR	7182	**428**
	IB-BGCRO-DR-CB ($p^{CB} = 15$)	**6934**	467
	IB-BGCRO-DR-CB ($p^{CB} = 10$)	6941	668
	IB-BGCRO-DR-CB ($p^{CB} = 5$)	6968	1312
	IB-BGCRO-DR-CB ($p^{CB} = 1$)	6966	6444
30	IB-BGCRO-DR	68262	**3932**
	IB-BGCRO-DR-CB ($p^{CB} = 15$)	**65364**	4303
	IB-BGCRO-DR-CB ($p^{CB} = 1$)	65823	60836

As in previous subsections, we note that this subspace expansion policy is also applicable to IB-BGMRES-DR. We refer the reader to [9, Figure G.1 and Table G.1 of Appendix G] for an illustration.

5.5. Behavior on sequences of slowly varying left-hand side problems. The example used in this section is from a finite element fracture mechanics problem in the field of fatigue and fracture of engineering components (denoted as the *FFEC* collection), which is fully documented in [16, section 4.1]. Over 2000 linear systems of size 3988×3988 from the *FFEC* collection need to be solved in order to capture the fracture progression, and among them 151 (linear systems 400–550) representing a typical subset of the fracture progression in which many cohesive elements break are examined in [16]. The solutions of these linear systems have been investigated using both GCRO-DR and GCROT (generalized conjugate residual with inner orthogonalization and outer truncation). We refer the reader to [8] for a comprehensive experimental analysis. For our numerical experiments we borrow the 10 linear systems numbered from 400 to 409 from the *FFEC* collection. For each set of linear systems we select the matrix and the corresponding right-hand sides that we expand to form a block of $p = 20$ by appending random linearly independent vectors.

We display the convergence histories for solving the first 3 consecutive families of such linear systems in the left plot of Figure 5. For the solution of the first linear system, the observations on the IB and DR mechanisms discussed in section 5.1 apply. Even though the coefficient matrix has changed, the recycling spectral information computed for the previous family still enables a faster convergence at the beginning of the solution of the next one. Specifically, for the solution of the first family the convergence histories of the two methods fully overlap until the first partial convergence occurs, as until this step the two methods are identical. From the initial slope of the subsequent families, it can be seen that the sequence of matrices are close enough to ensure that the recycled space from one system to the next is still beneficial to the convergence. The benefit of the partial convergence detection is also illustrated on that example since IB-BGCRO-DR still outperforms BGCRO-DR. The overall

benefit in term of *mvps* savings is illustrated in the right plot on a sequence of 10 linear systems, where the savings are more than 65% with respect to BGCRO-DR. Corresponding results are summarized in Table 4.

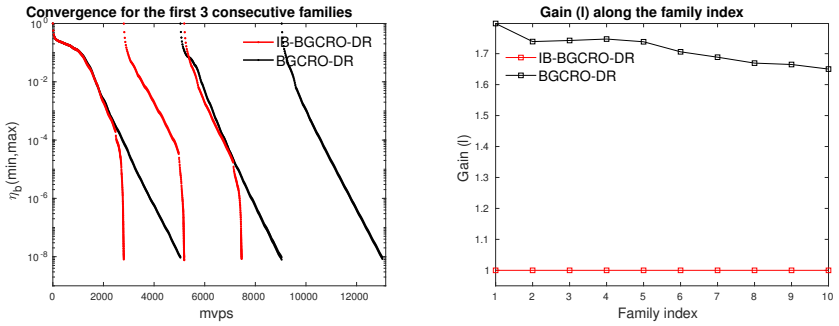

FIG. 5. *Convergence results of IB-BGCRO-DR and BGCRO-DR on a sequence of slowly changing left-hand sides described in section* 5.5*, where the coefficient matrices are built on the FFEC with* $p = 20$*,* $m_d = 300$*, and* $k = 15$*.*

TABLE 4
Numerical results in terms of mvps and its for section 5.5 *with* $p = 20$*,* $m_d = 300$*, and* $k = 15$*.*

Number of families	Method	*mvps*	*its*
3	BGCRO-DR	13050	651
	IB-BGCRO-DR	**7489**	**540**
10	BGCRO-DR	39935	1990
	IB-BGCRO-DR	**24200**	**1658**

5.6. A variant suited for flexible preconditioning. In this section, we illustrate the numerical behavior of the flexible variant IB-BFGCRO-DR that we have derived in section 2.5 and make a comparison with closely related variants, namely BFGCRO-DR (a straightforward block extension of FGCRO-DR [5]).

We consider a representative quantum chromodynamics (QCD) matrix from the University of Florida Sparse Matrix Collection [6]. It is the conf5.4-00l8x8-0500 matrix denoted as B_{QCD} of size 49152×49152 with the critical parameter $\kappa_c = 0.17865$ as a model problem. Thirty families of linear systems are constructed that are defined as $A^{(\ell)} = I - \kappa_c(\ell) B_{\mathrm{QCD}}$ with $0 \leq \kappa_c(\ell) < \kappa_c$ and $\ell = 1, 2, \ldots, 30$. We use the MATLAB function **linspace**$(0.1780, 0.1786, 30)$ to generate the parameters $\kappa_c(\ell)$ for a sequence of matrices and observe that those matrices have the same eigenvectors associated with shifted eigenvalues. A sequence of $p = 12$ successive canonical basis vectors are chosen to be the block of right-hand sides for a given left-hand side matrix following [16, section 4.3] so that the complete set of the right-hand sides for the ℓ linear systems reduces to the first $p \times \ell$ columns of the identity matrix. This choice could be supported by the fact that the problem of numerical simulations of QCD on a four-dimensional space-time lattice for solving QCD ab initio (cf. [16, section 4.3]) has a 12×12 block structure, and then a system with 12 right-hand sides related to a single lattice site is often of interest to solve.

The flexible preconditioner is defined by a 32-bit $ILU(0)$ factorization of the matrix involved in the linear system. In a 64-bit calculation framework, the preconditioning consists of casting the set of directions to be preconditioned in 32-bit format,

performing the forward/backward substitution in 32-bit calculation, and casting back the solutions in 64-bit arithmetic. The rounding applied to the vectors, cast from 64- to 32-bit format, has a nonlinear effect that makes the preconditioner nonlinear.

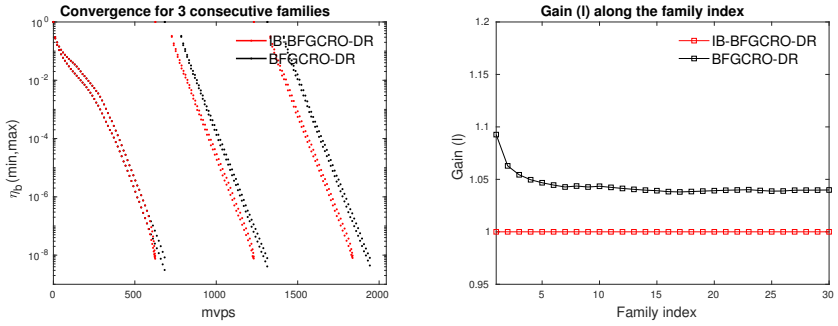

FIG. 6. *Behavior of the BGCRO-DR-solvers with flexible preconditioner on families of QCD matrices described in section* 5.6 *with* $p = 12$, $m_d = 180$, *and* $k = 90$. *Left: convergence histories of the largest/smallest backward errors* $\eta_{b^{(i)}}$ *at each mvps for* 3 *consecutive families. Right: gain (l) of the block methods with respect to IB-BFGCRO-DR along the family index.*

For those experiments, we attempt to favor the recycling of the space, because the matrices share the same invariant space, so that we choose a relatively large value for k that is $k = m_d/2$. We report in the left plot of Figure 6 the convergence histories of the two flexible block variants. Similarly to what has already been observed, the convergences are very similar on the first family and only differ when the partial convergence detection becomes active mostly in the last restart. For the second and third families, one can see that IB-BFGCRO-DR and BFGCRO-DR have identical convergence speeds. One can observe a shift in the convergence histories between the end of the solution of one family and the beginning of the next for both IB-BFGCRO-DR and BFGCRO-DR. This shift is due to the extra k *mvps* that have to be performed when the matrix changes in order to adapt the recycling space as follows:

1. Compute $A^{(\ell+1)}U_k^{(\ell)} = \tilde{C}_k$.
2. Compute the reduced QR-factorization of $\tilde{C}_k = C_k^{(\ell+1)}R$.
3. Update the basis of the deflation space $U_k^{(\ell+1)} = U_k^{(\ell)}R^{-1}$ so that $A^{(\ell+1)}U_k^{(\ell+1)} = C_k^{(\ell+1)}$.

Because k is large, we can clearly see this shift in the left plot of Figure 6. For this parameter selection in this section, it can be seen that the dominating effect on the convergence improvement is due to the space recycling and not the partial convergence detection. This observation is highlighted in the right plot of Figure 6, where the benefit of using IB-BFGCRO-DR rather than BFGCRO-DR diminishes when compared to previous experiments and is only about 4%. Numerical details are summarized in Table 5.

6. Concluding remarks. In this paper, we develop a new variant of the block GCRO-DR method, denoted as IB-BGCRO-DR, that inherits the appealing genes of its two parents [16, 19]. First, it inherits the capability of speeding up the convergence rate when solving sequences of linear systems by recycling spectral information from one family to the next. Second, the extended search space expansion policy enabled by the so-called partial convergence detection allows us to focus on the convergence by considering only the most important directions. Along this line, we introduce

TABLE 5
Numerical results in terms of mvps and its for section 5.6 *with* $p = 12$, $m_d = 15 \times p = 180$, *and* $k = 90$.

Number of families	Method	*mvps*	*its*
3	BFGCRO-DR	1944	**147**
	IB-BFGCRO-DR	**1838**	148
30	BFGCRO-DR	18774	**1347**
	IB-BFGCRO-DR	**18054**	1350

stopping-criterion driven search space expansion polices that enable us to ensure that a prescribed threshold used for the partial convergence detection will eventually lead to reaching a prescribed threshold for a backward error based stopping criterion. While introduced in the block GCRO context, those policies apply to any block minimum residual norm approach that relies on an Arnoldi-like relation and includes both block GMRES and GCRO variants. In exact arithmetic, these policies exploit the close link between the least squares residuals and the linear system residuals, which is guaranteed by the orthonormal basis of the residual space. Through numerical experiments, we show that the MGS re-orthogonalization between the columns of recycling space and initial block Arnoldi basis at restart combined with (BMGS ∘ HouseQR) in the block Arnoldi algorithm seems to generate a good enough orthonormal basis to ensure that such a property also holds in finite precision calculation. Following ideas from [14], future research could theoretically establish that this class of subspace augmentation algorithms is backward stable. To comply with mixed-precision calculation, the flexible preconditioning variant is also proposed, which would be of interest for emerging computing platforms where mixed-precision calculation could be a way to reduce data movement, which is foreseen as one of the major bottlenecks to reaching high performance.

Appendix A. Proof of Proposition 4.

Proof. From (4.1), (4.2), and (4.4), the initial residual block R_1 with partial convergence detection at restart could be described as

$$\begin{aligned}
R_1 &= [C_k, \mathbb{V}_1, P_0, \widetilde{W}_1][C_k, \mathbb{V}_1, P_0, \widetilde{W}_1]^H R_1 \\
&= [C_k, \mathbb{V}_1, P_0, \widetilde{W}_1][C_k, \mathbb{V}_1, P_0, \widetilde{W}_1]^H [\mathbb{V}_1^{new}, P_0^{new}]\hat{\Lambda}_1 \\
&= [C_k, \mathbb{V}_1, P_0, \widetilde{W}_1] \left([C_k, \mathbb{V}_1, P_0, \widetilde{W}_1]^H C_k R_{12} + [C_k, \mathbb{V}_1, P_0, \widetilde{W}_1]^H [\mathbb{V}_1, P_0] R_{22} \right) \hat{\Lambda}_1 \\
&= [C_k, \mathbb{V}_1, P_0, \widetilde{W}_1]\Lambda_1 \text{ with } \Lambda_1 = \begin{bmatrix} R_{12} \\ 0_{(p_1+p)\times p} \end{bmatrix} \hat{\Lambda}_1 + \begin{bmatrix} 0_{k\times p_1} & 0_{k\times q_1} \\ I_{p_1} & 0_{p_1\times q_1} \\ 0_{q_1\times p_1} & I_{q_1} \\ 0_{p_1\times p_1} & 0_{p_1\times q_1} \end{bmatrix} R_{22}\hat{\Lambda}_1
\end{aligned}$$

by $[\mathbb{V}_1^{new}, P_0^{new}] = C_k R_{12} + [\mathbb{V}_1, P_0] R_{22}$ obtained from (4.2). That can also be written as

$$\Lambda_1 = \begin{bmatrix} R_{12} \\ 0_{(p_1+p)\times p} \end{bmatrix} \hat{\Lambda}_1 + \begin{bmatrix} 0_{k\times p_1} & 0_{k\times q_1} \\ I_{p_1} & \Phi_1 \\ 0_{q_1\times p_1} & \\ 0_{p_1\times p_1} & 0_{p_1\times q_1} \end{bmatrix} R_{22}\hat{\Lambda}_1,$$

where

$$\Phi_1 = \begin{bmatrix} 0_{p_1\times q_1} \\ I_{q_1} \end{bmatrix} \in \mathbb{C}^{p\times q_1} \text{ and } q_1 + p_1 = p.$$

The right-hand sides of the least squares problem at iteration $(j+1)$ for $j = 1, 2, \ldots$ are defined by

$$
\begin{aligned}
\Lambda_{j+1} &= [C_k, \mathscr{V}_{j+1}, [P_j, \widetilde{W}_{j+1}]]^H R_1 = [C_k, \mathscr{V}_j, V_{j+1}, [P_j, \widetilde{W}_{j+1}]]^H R_1 \\
&= \left[C_k, \mathscr{V}_j, [P_{j-1}, \widetilde{W}_j]\mathbb{W}_1, [P_{j-1}, \widetilde{W}_j]\mathbb{W}_2, \widetilde{W}_{j+1}\right]^H R_1 \\
&= \left[C_k, \mathscr{V}_j, [P_{j-1}, \widetilde{W}_j][\mathbb{W}_1, \mathbb{W}_2], \widetilde{W}_{j+1}\right]^H [\mathbb{V}_1^{new}, P_0^{new}]\hat{\Lambda}_1 \\
&= \left(\left[C_k, \mathscr{V}_j, [P_{j-1}, \widetilde{W}_j][\mathbb{W}_1, \mathbb{W}_2], \widetilde{W}_{j+1}\right]^H C_k R_{12}\right. \\
&\qquad \left. + \left[C_k, \mathscr{V}_j, [P_{j-1}, \widetilde{W}_j][\mathbb{W}_1, \mathbb{W}_2], \widetilde{W}_{j+1}\right]^H [\mathbb{V}_1, P_0] R_{22}\right) \hat{\Lambda}_1 \\
&= \begin{bmatrix} R_{12} \\ 0_{(n_j+p+p_{j+1})\times p} \end{bmatrix} \hat{\Lambda}_1 \\
&\qquad + \begin{bmatrix} C_k^H \mathbb{V}_1 & C_k^H P_0 \\ \mathscr{V}_j^H \mathbb{V}_1 & \mathscr{V}_j^H P_0 \\ [V_{j+1}, P_j]^H \mathbb{V}_1 & [\mathbb{W}_1, \mathbb{W}_2]^H \left[P_{j-1}, \widetilde{W}_j\right]^H P_0 \\ \widetilde{W}_{j+1}^H \mathbb{V}_1 & \widetilde{W}_{j+1}^H P_0 \end{bmatrix} R_{22}\hat{\Lambda}_1 \\
&= \begin{bmatrix} R_{12} \\ 0_{(n_j+p+p_{j+1})\times p} \end{bmatrix} \hat{\Lambda}_1 \\
&\qquad + \begin{bmatrix} 0_{k\times p_1} & 0_{k\times q_1} \\ \begin{bmatrix} I_{p_1} \\ 0_{(n_j-p_1)\times p_1} \end{bmatrix} & \Phi_j(1:n_j, :) \\ 0_{p\times p_1} & [\mathbb{W}_1, \mathbb{W}_2]^H \begin{bmatrix} P_{j-1}^H \\ \widetilde{W}_j^H \end{bmatrix} P_0 \\ 0_{p_{j+1}\times p_1} & 0_{p_{j+1}\times q_1} \end{bmatrix} R_{22}\hat{\Lambda}_1 \\
&= \begin{bmatrix} R_{12} \\ 0_{(n_j+p+p_{j+1})\times p} \end{bmatrix} \hat{\Lambda}_1 \\
&\qquad + \begin{bmatrix} 0_{k\times p_1} & 0_{k\times q_1} \\ \begin{bmatrix} I_{p_1} \\ 0_{(n_j-p_1)\times p_1} \end{bmatrix} & \Phi_j(1:n_j, :) \\ 0_{p\times p_1} & [\mathbb{W}_1, \mathbb{W}_2]^H \begin{bmatrix} \Phi_j(n_j+1:n_j+q_j, :) \\ 0_{p_j\times q_1} \end{bmatrix} \\ 0_{p_{j+1}\times p_1} & 0_{p_{j+1}\times q_1} \end{bmatrix} R_{22}\hat{\Lambda}_1 \\
&= \begin{bmatrix} R_{12} \\ 0_{(n_j+p+p_{j+1})\times p} \end{bmatrix} \hat{\Lambda}_1 + \begin{bmatrix} 0_{k\times p_1} & 0_{k\times q_1} \\ \begin{bmatrix} I_{p_1} \\ 0_{(n_j+p-p_1)\times p_1} \end{bmatrix} & \Phi_{j+1} \\ 0_{p_{j+1}\times p_1} & 0_{p_{j+1}\times q_1} \end{bmatrix} R_{22}\hat{\Lambda}_1,
\end{aligned}
$$

where $\Phi_{j+1} \in \mathbb{C}^{(n_j+p)\times q_1}$ for $j = 1, 2, \ldots$. ∎

Appendix B. The SVD decomposition of the least squares residual and the solution of the least squares problem. The partial convergence detection mechanism allows us to extract from the residual spaces new directions to expand the search space at the next iteration of the block method. The selection consists of extracting the directions that contribute the most to the scaled residual block and is based on the SVD of the scaled least squares residual. In this section, we detail

how the solution of the least squares problem (2.11) enables us to compute easily and cheaply the SVD of the associated scaled (least squares) residual block. The least squares problem

$$Y_j = \underset{Y \in \mathbb{C}^{(k+n_j)\times p}}{\operatorname{argmin}} \left\| \Lambda_j - \underline{\mathscr{F}}_j Y \right\|_F, \text{ with } \underline{\mathscr{F}}_j \in \mathbb{C}^{(k+n_j+p)\times(k+n_j)}, \tag{B.1}$$

is solved by using a full QR-factorization of $\underline{\mathscr{F}}_j = Q_j^{LS} R_j^{LS}$, where the superscript ^{LS} stands for least squares, $Q_j^{LS} = [Q_j^{LS(1)},\ Q_j^{LS(2)}]$ with $Q_j^{LS(1)} \in \mathbb{C}^{(k+n_j+p)\times(k+n_j)}$ and $Q_j^{LS(2)} \in \mathbb{C}^{(k+n_j+p)\times p}$, and $R_j^{LS} = \begin{bmatrix} R_j^{LS(1)} \\ 0_{p\times(k+n_j)} \end{bmatrix} \in \mathbb{C}^{(k+n_j+p)\times(k+n_j)}$ with $R_j^{LS(1)} \in \mathbb{C}^{(k+n_j)\times(k+n_j)}$ is an upper triangular matrix, from which the reduced QR-factorization of $\underline{\mathscr{F}}_j$ is formulated as $\underline{\mathscr{F}}_j = Q_j^{LS(1)} R_j^{LS(1)}$ if $Q_j^{LS(1)}$ is considered as an orthogonal basis of $\underline{\mathscr{F}}_j$. Thus, we could still formulate Y_j in a relatively economic way as

$$Y_j = (R_j^{LS(1)})^{-1}((Q_j^{LS(1)})^H \Lambda_j) \in \mathbb{C}^{(k+n_j)\times p}, \tag{B.2}$$

from which we could deduce the residual of the least squares problem described in (3.6) as follows:

$$\begin{aligned}
\Lambda_j - \underline{\mathscr{F}}_j Y_j &= \Lambda_j - Q_j^{LS} R_j^{LS} Y_j = Q_j^{LS} \left((Q_j^{LS})^H \Lambda_j - R_j^{LS} Y_j \right), \\
&= Q_j^{LS} \left(\begin{bmatrix} (Q_j^{LS(1)})^H \\ (Q_j^{LS(2)})^H \end{bmatrix} \Lambda_j - \begin{bmatrix} R_j^{LS(1)} \\ 0_{p\times(k+n_j)} \end{bmatrix} Y_j \right), \\
&= Q_j^{LS} \left(\begin{bmatrix} 0_{(k+n_j)\times(k+n_j+p)} \\ (Q_j^{LS(2)})^H \end{bmatrix} \Lambda_j \right), \\
&= Q_j^{LS} \begin{pmatrix} 0_{(k+n_j)\times p} \\ R_j^{\ell s} \end{pmatrix},
\end{aligned}$$

where $R_j^{\ell s} = (Q_j^{LS(2)})^H \Lambda_j \in \mathbb{C}^{p\times p}$ are the last p rows of $(Q_j^{LS})^H \Lambda_j$. The SVD of scaled residual $R_j^{\ell s} D_\varepsilon$ can be written as

$$R_j^{\ell s} D_\varepsilon = U_{\ell s} \Sigma V_{\ell s}^H,$$

so that the SVD of the scaled least squares residual is

$$\left(\Lambda_j - \underline{\mathscr{F}}_j Y_j\right) D_\varepsilon = \underbrace{Q_j^{LS} \begin{pmatrix} 0_{(n_j+k)\times p} & I_{n_j+k} \\ U_{\ell s} & 0_{p\times(n_j+k)} \end{pmatrix}}_{\text{Unitary}} \begin{pmatrix} \Sigma \\ 0_{(n_j+k)\times p} \end{pmatrix} V_{\ell s}^H.$$

Acknowledgments. We would like to thank Matthieu Simonin (who developed a C++ implementation of the solver—available at https://gitlab.inria.fr/solverstack/fabulous/—in the framework of the PRACE 6IP project) for his reading and comments on an earlier version of this document. We are extremely grateful to the Associate Editor and the anonymous referees for the stimulating and constructive exchanges during the review process. Their comments and questions helped us to not only improve the readability of the article but also enrich its scientific content.

REFERENCES

[1] E. Agullo, L. Giraud, and Y.-F. Jing, *Block GMRES method with inexact breakdowns and deflated restarting*, SIAM J. Matrix Anal. Appl., 35 (2014), pp. 1625–1651, https://doi.org/10.1137/140961912.

[2] H. Calandra, S. Gratton, R. Lago, X. Vasseur, and L. M. Carvalho, *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 (2013), pp. S345–S367, https://doi.org/10.1137/120883037.

[3] E. Carson, K. Lund, M. Rozloznik, and S. Thomas, *Block Gram-Schmidt algorithms and their stability properties*, Linear Algebra Appl., 638 (2022), pp. 150–195.

[4] L. M. Carvalho, S. Gratton, R. Lago, and X. Vasseur, *A Flexible Generalized Conjugate Residual Method with Inner Orthogonalization and Deflated Restarting*, Tech. Report TR/PA/10/10, CERFACS, Toulouse, France, 2010.

[5] L. M. Carvalho, S. Gratton, R. Lago, and X. Vasseur, *A flexible generalized conjugate residual method with inner orthogonalization and deflated restarting*, SIAM J. Matrix Anal. Appl., 32 (2011), pp. 1212–1235, https://doi.org/10.1137/100786253.

[6] T. A. Davis and Y. Hu, *The University of Florida Sparse Matrix Collection*, ACM Trans. Math. Softw., 38 (2011), 1.

[7] E. de Sturler, *Nested Krylov methods based on GCR*, J. Comput. Appl. Math., 67 (1996), pp. 15–41.

[8] E. de Sturler, *Truncation strategies for optimal Krylov subspace methods*, SIAM J. Numer. Anal., 36 (1999), pp. 864–889, https://doi.org/10.1137/S0036142997315950.

[9] L. Giraud, Y.-F. Jing, and Y.-F. Xiang, *A Block Minimum Residual Norm Subspace Solver with Partial Convergence Management for Sequences of Linear Systems*, Research Report 9393, Inria, Bordeaux Sud-Ouest, 2021.

[10] M. H. Gutknecht, *Block Krylov space methods for linear systems with multiple right-hand sides: An introduction*, in Modern Mathematical Models, Methods and Algorithms for Real World Systems, I. S. Duff, A. H. Siddiqi, and O. Christensen, eds., Anamaya Publishers, New Delhi, India, 2006, pp. 420–447.

[11] R. B. Morgan, *A restarted GMRES method augmented with eigenvectors*, SIAM J. Matrix Anal. Appl., 16 (1995), pp. 1154–1171, https://doi.org/10.1137/S0895479893253975.

[12] R. B. Morgan, *GMRES with deflated restarting*, SIAM J. Sci. Comput., 24 (2002), pp. 20–37.

[13] R. B. Morgan, *Restarted block GMRES with deflation of eigenvalues*, Appl. Numer. Math., 54 (2005), pp. 222–236, https://doi.org/10.1137/S1064827599364659.

[14] C. C. Paige, M. Rozložník, and Z. Strakoš, *Modified Gram–Schmidt (MGS), least squares, and backward stability of MGS-GMRES*, SIAM J. Matrix Anal. Appl., 28 (2006), pp. 264–284, https://doi.org/10.1137/050630416.

[15] M. L. Parks, *The Iterative Solution of a Sequence of Linear Systems Arising from Nonlinear Finite Element Analysis*, Ph.D. dissertation UIUCDCS-R-2005-2497, University of Illinois at Urbana-Champaign, 2005.

[16] M. L. Parks, E. de Sturler, G. Mackey, D. D. Johnson, and S. Maiti, *Recycling Krylov subspaces for sequences of linear systems*, SIAM J. Sci. Comput., 28 (2006), pp. 1651–1674, https://doi.org/10.1137/040607277.

[17] M. L. Parks, K. M. Soodhalter, and D. B. Szyld, *A Block Recycled GMRES Method with Investigations into Aspects of Solver Performance*, preprint, http://arxiv.org/abs/1604.01713, 2016.

[18] L. G. Ramos, R. Kehl, and R. Nabben, *Projections, deflation, and multigrid for nonsymmetric matrices*, SIAM J. Matrix Anal. Appl., 41 (2020), pp. 83–105, https://doi.org/10.1137/18M1180268.

[19] M. Robbé and M. Sadkane, *Exact and inexact breakdowns in the block GMRES method*, Linear Algebra Appl., 419 (2006), pp. 265–285.

[20] Y. Saad and M. H. Schultz, *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Stat. Comput., 7 (1986), pp. 856–869, https://doi.org/10.1137/0907058.

[21] K. Soodhalter, *Krylov Subspace Methods with Fixed Memory Requirements: Nearly Hermitian Linear Systems and Subspace Recycling*, Ph.D. dissertation, Temple University, Philadelphia, 2012.

[22] D.-L. Sun, B. Carpentieri, T.-Z. Huang, and Y.-F. Jing, *A spectrally preconditioned and initially deflated variant of the restarted block GMRES method for solving multiple right-hand sides linear systems*, Internat. J. Mech. Sci., 144 (2018), pp. 775–787.

[23] D.-L. Sun, T.-Z. Huang, B. Carpentieri, and Y.-F. Jing, *Flexible and deflated variants of*

the block shifted GMRES method, J. Comput. Appl. Math., 345 (2019), pp. 168–183.

[24] D.-L. Sun, T.-Z. Huang, B. Carpentieri, and Y.-F. Jing, *A new shifted block GMRES method with inexact breakdowns for solving multi-shifted and multiple right-hand sides linear systems*, J. Sci. Comput., 78 (2019), pp. 746–769.

[25] D.-L. Sun, T.-Z. Huang, Y.-F. Jing, and B. Carpentieri, *A block GMRES method with deflated restarting for solving linear systems with multiple shifts and multiple right-hand sides*, Numer. Linear Algebra Appl., 25 (2018), e2148, https://doi.org/10.1002/nla.2148.

[26] A. Tajaddini, G. Wu, F. Saberi-Movahed, and N. Azizizadeh, *Two new variants of the simpler block GMRES method with vector deflation and eigenvalue deflation for multiple linear systems*, J. Sci. Comput., 86 (2021), 9.

[27] Y.-F. Xiang, Y.-F. Jing, and T.-Z. Huang, *A new projected variant of the deflated block conjugate gradient method*, J. Sci. Comput., 80 (2019), pp. 1116–1138.

[28] F. Xue and H. C. Elman, *Fast inexact subspace iteration for generalized eigenvalue problems with spectral transformation*, Linear Algebra Appl., 435 (2011), pp. 601–622.