SIAM J. SCI. COMPUT.
Vol. 35, No. 5, pp. S345–S367

A MODIFIED BLOCK FLEXIBLE GMRES METHOD WITH DEFLATION AT EACH ITERATION FOR THE SOLUTION OF NON-HERMITIAN LINEAR SYSTEMS WITH MULTIPLE RIGHT-HAND SIDES*

HENRI CALANDRA†, SERGE GRATTON‡, RAFAEL LAGO§, XAVIER VASSEUR¶, AND LUIZ MARIANO CARVALHO‖

Abstract. We propose a variant of the block GMRES method for the solution of linear systems of equations with multiple right-hand sides. We investigate a deflation strategy to detect when a linear combination of approximate solutions is already known that avoids performing expensive computational operations with the system matrix. This is especially useful when the cost of the preconditioner is supposed to be larger than the cost of orthogonalization in the block Arnoldi procedure. We specifically focus on the block GMRES method incorporating deflation at the end of each iteration proposed by Robbé and Sadkane [M. Robbé and M. Sadkane, *Linear Algebra Appl.*, 419 (2006), pp. 265–285]. We extend their contribution by proposing that deflation be performed also at the beginning of each cycle. This change leads to a modified least-squares problem to be solved at each iteration and gives rise to a different behavior especially when multiple restarts are required to reach convergence. Additionally we investigate truncation techniques, aiming at reducing the computational cost of the iteration. This is particularly useful when the number of right-hand sides is large. Finally, we address the case of variable preconditioning, an important feature when iterative methods are used as preconditioners, as investigated here. The numerical experiments performed in a parallel environment show the relevance of the proposed variant on a challenging application related to geophysics. A savings of up to 35% in terms of computational time—at the same memory cost—is obtained with respect to the original method on this application.

Key words. block Krylov space method, block size reduction, deflation at each iteration, flexible preconditioning, multiple right-hand sides

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

DOI. 10.1137/120883037

1. Introduction.

We consider block Krylov space methods for the solution of linear systems of equations with p right-hand sides given at once of the form $AX = B$, where $A \in \mathbb{C}^{n\times n}$ is supposed to be a nonsingular non-Hermitian matrix, $B \in \mathbb{C}^{n\times p}$ is supposed to be full rank, and $X \in \mathbb{C}^{n\times p}$. Although the number of right-hand sides p might be relatively large, we suppose here that the dimension of the problem n is always much larger. Later, we denote by $X_0 \in \mathbb{C}^{n\times p}$ the initial block iterate, and by $R_0 = B - AX_0$ the initial block residual. As stated in [25, 26] a block Krylov space method for solving the p systems is an iterative method that generates approximations $X_m \in \mathbb{C}^{n\times p}$ with $m \in \mathbb{N}$ such that

$$X_m - X_0 \in \mathcal{K}_m^{\square}(A, R_0),$$

*Received by the editors June 29, 2012; accepted for publication (in revised form) June 25, 2013; published electronically October 28, 2013.
http://www.siam.org/journals/sisc/35-5/88303.html

†TOTAL, Centre Scientifique et Technique Jean Féger, F-64000 Pau, France (Henri.Calandra@total.com).

‡INPT-IRIT, University of Toulouse, and ENSEEIHT, BP 7122, F-31071 Toulouse Cedex 7, France (Serge.Gratton@enseeiht.fr).

§CERFACS, F-31057 Toulouse Cedex 1, France (lago@cerfacs.fr).

¶CERFACS and HiePACS project joint INRIA-CERFACS Laboratory, F-31057 Toulouse Cedex 1, France (vasseur@cerfacs.fr).

‖Applied Mathematics Department, IME-UERJ, 20559-900, Rio de Janeiro, RJ, Brazil (luizmc@gmail.com).

where the block Krylov space $K_m^{\square}(A, R_0)$ (in the nonpreconditioned case) is defined as

$$\mathcal{K}_m^{\square}(A, R_0) = \left\{ \sum_{k=0}^{m-1} A^k R_0 \gamma_k \ \forall \gamma_k \in \mathbb{C}^{p\times p}, \text{ with } k \mid 0 \le k \le m-1 \right\} \subset \mathbb{C}^{n\times p}.$$

We refer the reader to [25] for a recent detailed overview on block Krylov subspace methods and note that most of the standard Krylov subspace methods have a block counterpart (see, e.g., block GMRES [48], block BiCGStab [24], block IDR(s) [16], and block QMR [22]). In this paper we mainly focus on restarted block Krylov subspace methods that satisfy a minimum norm property as introduced in [42, section 6.12].

Block Krylov subspace methods are increasingly popular in many application areas in computational science and engineering (e.g., electromagnetic scattering (monostatic radar cross section analysis) [10, 31, 44], lattice quantum chromodynamics [43], model reduction in circuit simulation [21], stochastic finite element with uncertainty restricted to the right-hand side [18], and sensitivity analysis of mechanical systems [7]). To be effective in terms of computational operations it is recognized that these methods must incorporate a strategy for detecting when a linear combination of the systems has approximately converged [25]. This explicit block size reduction is called deflation, as discussed in [25]. First, a simple strategy to remove useless information from a block Krylov subspace—called initial deflation—consists of detecting possible linear dependency in the block right-hand side B or in the initial block residual R_0 [25, section 12], [31, section 3.7.2]. When a restarted block Krylov subspace method is used, this block size reduction can be also performed at each initial computation of the block residual, i.e., at the beginning of each cycle [25, section 14]. In addition Arnoldi deflation [25] may be considered; it aims at detecting a near rank deficiency occurring in the block Arnoldi procedure to later reduce the current block size. These three strategies based on rank-revealing QR-factorizations [11] or singular value decompositions [23] have been notably proposed both in the Hermitian [35, 40] and non-Hermitian [1, 4, 14, 22, 33, 36] cases for block Lanczos methods. They have been shown to be effective with respect to standard block Krylov subspace methods. While initial deflation or deflation at the beginning of a cycle are currently popular, block Krylov subspace methods based on a norm minimization property incorporating deflation at each iteration have rarely been studied (see, e.g., [7] for a discussion on deflated block Arnoldi methods).

In this paper we focus only on block GMRES based methods [48] and refer the reader to [7, 22, 33, 34] for advanced block Lanczos methods with deflation. In [39] Robbé and Sadkane introduced the notion of inexact breakdown to study block size reduction techniques in block GMRES. Two criteria have been proposed, based either on the numerical rank of the generated block Krylov basis (W-criterion) or on the numerical rank of the block residual (R-criterion). Numerical experiments on academic problems of small dimension with a reduced number of right-hand sides illustrated the advantages and drawbacks of each variant versus standard block GMRES. Further numerical experiments can be found in [29]. Another method relying on such a strategy is the dynamic BGMRES (DBGMRES) [15], which is an extension of block loose GMRES [5]. We also refer the reader to [7], where deflated block Arnoldi methods, in addition to Lanczos, are discussed on a real application problem in structural mechanics. The combination of block GMRES performing deflation at each iteration and variable preconditioning has rarely been addressed in the literature. Variable preconditioning is often required when solving large linear systems of equations. This

is notably the case when inexact solutions of the preconditioning system using, e.g., nonlinear smoothers in multigrid [37] or approximate interior solvers in domain decomposition methods [46, section 4.3] are considered. Thus the main purpose of the paper is to derive a flexible minimal norm block Krylov subspace method that incorporates block size reduction at each iteration suited to the solution of large-scale linear systems (where expensive variable preconditioners are often used) with possibly a large number of right-hand sides. This is especially useful when the cost of the preconditioner is supposed to be larger than the cost of orthogonalization in the block Arnoldi procedure.

The paper is organized as follows. First we will introduce in section 2 the block GMRES method with deflation at each iteration proposed in [39], since it will constitute the basis for further developments. We will notably describe how deflation at each iteration is performed. In section 3 we first explain the main motivations for deriving the proposed variant and analyze its main mathematical properties. Algorithmic details are then presented in section 4 together with an analysis of the computational cost and memory requirements. Then in section 5 we demonstrate the effectiveness of the proposed algorithm on an application related to geophysics. Finally, we draw some conclusions in section 6.

2. Block GMRES with deflation at each iteration. In this section we review the block GMRES method with deflation at each iteration (later denoted BGMRES-R[1]) [39] for the solution of linear systems with a non-Hermitian matrix and multiple right-hand sides given at once. We first introduce notation used in the manuscript and then describe the main mathematical properties of BGMRES-R.

2.1. Notation. Throughout this paper we denote by $\|.\|_2$ the Euclidean norm, by $\|.\|_F$ the Frobenius norm, by $I_k \in \mathbb{C}^{k\times k}$ the identity matrix of dimension k, and by $0_{i\times j} \in \mathbb{C}^{i\times j}$ the zero rectangular matrix with i rows and j columns. The superscript ^{H} denotes the transpose conjugate operation. Given a vector $d \in \mathbb{C}^k$ with components d_i, $D = \mathrm{diag}(d_1, \ldots, d_k)$ is the diagonal matrix $D \in \mathbb{C}^{k\times k}$ such that $D_{ii} = d_i$. If $C \in \mathbb{C}^{k\times l}$, we denote the singular values of C by $\sigma_1(C) \geq \cdots \geq \sigma_{\min(k,l)}(C) \geq 0$. Finally, $e_m \in \mathbb{C}^n$ denotes the mth canonical basis vector of $\mathbb{C}^n$. In describing our algorithms (Algorithms 1–4), we adopt notation similar to that of MATLAB. For instance, $U(i, j)$ denotes the U_{ij} entry of matrix U, $U(1 : m, 1 : j)$ refers to the submatrix made of the first m rows and first j columns of U, and $U(:, j)$ corresponds to its jth column.

2.2. Overview. Next we provide a brief overview of the block GMRES method with deflation at each iteration, introduced in [39], and specifically focus on the variant with a block size reduction strategy based on the numerical rank of the block residual (R-criterion [39, section 4]). More precisely we propose to analyze a given cycle of this method in the next subsections. Throughout the paper we denote by $X_0 \in \mathbb{C}^{n\times p}$ the current approximation of the solution, and by $R_0 \in \mathbb{C}^{n\times p}$ the corresponding true block residual ($R_0 = B - AX_0$), both obtained at the beginning of the cycle that we consider. $D \in \mathbb{C}^{p\times p}$ represents a nonsingular diagonal scaling matrix defined as $D = \mathrm{diag}(b_1, \ldots, b_p)$ with $b_l = ||B(:, l)||_2$, $1 \leq l \leq p$. Finally, we assume that the QR factorization of R_0D^{-1} has been performed as

$$R_0 D^{-1} = \hat{\mathcal{V}}_1 \hat{\Lambda}_0, \tag{2.1}$$

[1]The suffix "R" is used to emphasize that we exclusively consider the block GMRES method with deflation at each iteration based on the R-criterion proposed by Robbé and Sadkane in [39].

with $\hat{\mathcal{V}}_1 \in \mathbb{C}^{n\times p}$ having orthonormal columns and $\hat{\Lambda}_0 \in \mathbb{C}^{p\times p}$ assuming[2] $\operatorname{rank}(R_0 D^{-1}) = p$. R_0 ($R_0 D^{-1}$) is named the initial block residual (respectively, scaled initial block residual), where the term "initial" refers to the beginning of the cycle that we consider.

2.2.1. Deflated Arnoldi relation. If $K \in \mathbb{C}^{n\times p}$ denotes a matrix with orthonormal columns containing all the p new Krylov directions at iteration $j-1$, the most expensive part of the algorithm at the jth iteration lies in the p applications of the variable preconditioner supposed to be expensive. To be effective in terms of computational operations it is widely recognized that block Krylov subspace methods must rely on a strategy for detecting when a linear combination of the systems has approximately converged [25, 31]. In the framework of block Krylov subspace methods based on a norm minimization property, Robbé and Sadkane [39] have first proposed a block GMRES algorithm that relies on deflation at each iteration of a given cycle. To do so, they have introduced a modified version of the block Arnoldi algorithm—later called *deflated block Arnoldi*—in which $\operatorname{range}(K)$ is judiciously decomposed into

$$\operatorname{range}(K) = \operatorname{range}(V_j) \oplus \operatorname{range}(P_{j-1}), \quad \text{with } \begin{bmatrix} V_j & P_{j-1} \end{bmatrix}^H \begin{bmatrix} V_j & P_{j-1} \end{bmatrix} = I_p, \tag{2.2}$$

where $V_j \in \mathbb{C}^{n\times k_j}$, $P_{j-1} \in \mathbb{C}^{n\times d_j}$ with $k_j + d_j = p$. In other words, k_j Krylov directions are effectively considered at iteration j, while d_j directions are left aside (or deflated) at the same iteration. We note that literally the "best" subspace of $\operatorname{range}(K)$ of dimension k_j is chosen (not just k_j columns of K) defining V_j, leaving the remaining subspace in $\operatorname{range}(P_{j-1})$ (i.e., the deflated subspace is spanned by $\operatorname{range}(P_{j-1})$ at iteration j). Based on this decomposition, the deflated orthonormalization procedure will apply preconditioning and matrix-vector products *only* over the chosen k_j directions of V_j. Next we briefly describe the jth iteration of the resulting method.

Defining $s_0 = 0$, $s_j = s_{j-1} + k_j$ and given $\begin{bmatrix} \mathcal{V}_j & P_{j-1} \end{bmatrix} \in \mathbb{C}^{n\times(s_j+d_j)}$ with orthonormal columns, the following block Arnoldi relation is assumed to hold at the beginning of the jth iteration of the deflated block Arnoldi procedure ($j > 1$):

$$A\mathcal{V}_{j-1} = \begin{bmatrix} \mathcal{V}_j & P_{j-1} \end{bmatrix} \mathcal{H}_{j-1}, \tag{2.3}$$

with $\mathcal{V}_{j-1} \in \mathbb{C}^{n\times s_{j-1}}$, $\mathcal{V}_j \in \mathbb{C}^{n\times s_j}$, $P_{j-1} \in \mathbb{C}^{n\times d_j}$, and $\mathcal{H}_{j-1} \in \mathbb{C}^{(s_{j-1}+p)\times s_{j-1}}$. The jth iteration of the deflated block Arnoldi procedure produces matrices $\hat{V}_{j+1} \in \mathbb{C}^{n\times k_j}$, $\hat{\mathcal{H}}_j \in \mathbb{C}^{(s_j+p)\times s_j}$ which satisfy

$$A \begin{bmatrix} \mathcal{V}_{j-1} & V_j \end{bmatrix} = \begin{bmatrix} \mathcal{V}_j & P_{j-1} & \hat{V}_{j+1} \end{bmatrix} \hat{\mathcal{H}}_j, \tag{2.4}$$

where $\hat{\mathcal{H}}_j$ has the following block structure:

$$\hat{\mathcal{H}}_j = \left[\begin{array}{c|c} \mathcal{H}_{j-1} & H_j \\ \hline 0_{k_j\times s_{j-1}} & H_{j+1,j} \end{array}\right],$$

with $H_j \in \mathbb{C}^{(s_{j-1}+p)\times k_j}$ and $H_{j+1,j} \in \mathbb{C}^{k_j\times k_j}$ (see Algorithm 1 for a complete description of this iteration). We assume that $\hat{\mathcal{H}}_j$ is always full rank; i.e., no Arnoldi breakdown occurs. We note that Arnoldi breakdowns rarely happen in practice (see, e.g.,

[2]The situation of $R_0 D^{-1}$ being rank-deficient in exact arithmetic is often referred to as *initial breakdown* [25]. However, as in [39], for the sake of simplicity we consider that $\operatorname{rank}(R_0 D^{-1}) = p$ holds at each cycle. We refer the reader to [25] for details on how to work around initial deflation, and we point out that this phenomenon has not been observed in our numerical experiments.

[25, section 13]). Therefore the possibility of an Arnoldi breakdown has not been considered in this paper, as in recent contributions [13, 25, 39]. Defining $\hat{\mathcal{V}}_{j+1} \in \mathbb{C}^{n\times(s_j+p)}$ as

$$\hat{\mathcal{V}}_{j+1} = \begin{bmatrix}\mathcal{V}_j & P_{j-1} & \hat{V}_{j+1}\end{bmatrix}, \tag{2.5}$$

the block Arnoldi relation (2.4) can then be stated as

$$A\mathcal{V}_j = \hat{\mathcal{V}}_{j+1}\hat{\mathcal{H}}_j. \tag{2.6}$$

Next the key idea is to perform the subspace decomposition previously mentioned in (2.2) as

$$\begin{aligned}\begin{bmatrix}\mathcal{V}_j & V_{j+1} & P_j\end{bmatrix} &= \begin{bmatrix}\mathcal{V}_j & P_{j-1} & \hat{V}_{j+1}\end{bmatrix}\mathcal{F}_{j+1},\\ \begin{bmatrix}\mathcal{V}_{j+1} & P_j\end{bmatrix} &= \hat{\mathcal{V}}_{j+1}\mathcal{F}_{j+1},\end{aligned} \tag{2.7}$$

where $\mathcal{F}_{j+1} \in \mathbb{C}^{(s_j+p)\times(s_j+p)}$ is a unitary matrix. We address how to determine $\mathcal{F}_{j+1}$ later in section 2.2.4. Hence we obtain

$$A\mathcal{V}_j = \hat{\mathcal{V}}_{j+1}\mathcal{F}_{j+1}\mathcal{F}_{j+1}^H\hat{\mathcal{H}}_j.$$

Defining $\mathcal{H}_j \in \mathbb{C}^{(s_j+p)\times s_j}$ as $\mathcal{H}_j = \mathcal{F}_{j+1}^H\hat{\mathcal{H}}_j$, we then deduce (since $\mathcal{F}_{j+1}$ is unitary)

$$A\mathcal{V}_j = \begin{bmatrix}\mathcal{V}_{j+1} & P_j\end{bmatrix}\mathcal{H}_j,$$

which is precisely the block Arnoldi relation required at the beginning of the next iteration (compare with relation (2.3)). This last relation can be written as

$$A\mathcal{V}_j = \begin{bmatrix}\mathcal{V}_{j+1} & P_j\end{bmatrix}\begin{bmatrix}\mathcal{L}_j\\ G_j\end{bmatrix},$$

where $\mathcal{L}_j$ corresponds to the $(s_j + k_{j+1}) \times s_j$ upper part of $\mathcal{H}_j$, and G_j to the $d_{j+1} \times s_j$ lower part of $\mathcal{H}_j$. This is exactly the core relation proposed in [39, section 5, Algorithm 2].

2.2.2. Representation of the scaled initial block residual. At the beginning of the cycle the initial subspace decomposition is supposed to hold in BGMRES-R:

$$\mathcal{V}_1 = \hat{\mathcal{V}}_1. \tag{2.8}$$

Consequently p Krylov directions are effectively considered at the first iteration of a given cycle ($k_1 = p$), while no directions are deflated at the same iteration ($d_1 = 0$). At iteration j of the cycle ($1 \le j \le m$), we define the quantity $\hat{\Lambda}_j \in \mathbb{C}^{(s_j+p)\times p}$ as

$$\hat{\Lambda}_j = \begin{bmatrix}\hat{\Lambda}_0\\ 0_{s_j\times p}\end{bmatrix}. \tag{2.9}$$

It is then straightforward to prove that R_0D^{-1} can be written as

$$R_0D^{-1} = \hat{\mathcal{V}}_{j+1}\hat{\Lambda}_j, \tag{2.10}$$

which means that $\hat{\Lambda}_j$ is the reduced representation of the scaled initial block residual in the $\hat{\mathcal{V}}_{j+1}$ basis.

2.2.3. Minimization property. We denote by $Y_j \in \mathbb{C}^{s_j \times p}$ the solution of the reduced minimization problem $\mathcal{P}_r$ considered in BGMRES-R:

$$\mathcal{P}_r : Y_j = \underset{Y \in \mathbb{C}^{s_j \times p}}{\operatorname{argmin}} ||\hat{\Lambda}_j - \hat{\mathcal{H}}_j Y||_F, \tag{2.11}$$

with $\hat{\mathcal{H}}_j$ and $\hat{\Lambda}_j$ defined in (2.6) and (2.9), respectively. We also denote by $\hat{\mathcal{R}}_j \in \mathbb{C}^{(s_j+p)\times p}$ the block residual of the reduced least-squares problem $\mathcal{P}_r$, i.e., $\hat{\mathcal{R}}_j = \hat{\Lambda}_j - \hat{\mathcal{H}}_j Y_j$ $(1 \le j \le m)$, and define $\hat{\mathcal{R}}_0 \in \mathbb{C}^{p\times p}$ as $\hat{\mathcal{R}}_0 = \hat{\Lambda}_0$. We recall in Proposition 2.1 the norm minimization property occurring in BGMRES-R.

PROPOSITION 2.1. *In block GMRES with deflation at each iteration (BGMRES-R), solving the reduced minimization problem $\mathcal{P}_r$ of* (2.11) *amounts to minimizing the Frobenius norm of the block true residual $||B-AX||_F$ over the space $X_0+\text{range}(\mathcal{V}_j Y D)$ at iteration j $(1 \le j \le m)$ of a given cycle, i.e.,*

$$\begin{aligned} \underset{Y \in \mathbb{C}^{s_j \times p}}{\operatorname{argmin}} ||\hat{\Lambda}_j - \hat{\mathcal{H}}_j Y||_F &= \underset{Y \in \mathbb{C}^{s_j \times p}}{\operatorname{argmin}} ||R_0 D^{-1} - A\mathcal{V}_j Y||_F \\ &= \underset{Y \in \mathbb{C}^{s_j \times p}}{\operatorname{argmin}} ||B - A(X_0 + \mathcal{V}_j Y D)||_F, \end{aligned} \tag{2.12}$$

with $\hat{\mathcal{H}}_j$ and $\hat{\Lambda}_j$ defined in (2.6) *and* (2.9)*, respectively.*

Proof. Due to relations (2.4) and (2.10), $||R_0 D^{-1} - A\mathcal{V}_j Y||_F$ can be written as

$$||R_0 D^{-1} - A\mathcal{V}_j Y||_F = \|\hat{\mathcal{V}}_{j+1}(\hat{\Lambda}_j - \mathcal{H}_j Y)\|_F.$$

Since $\mathcal{V}_{j+1}$ has orthonormal columns and since the Frobenius norm is unitarily invariant, the last equality becomes

$$||R_0 D^{-1} - A\mathcal{V}_j Y||_F = \|\hat{\Lambda}_j - \hat{\mathcal{H}}_j Y\|_F.$$

D being a diagonal matrix, the relation (2.12) is then due to elementary properties of the Frobenius norm; namely, the squared Frobenius norm of a matrix is the sum of the squares of the Euclidean norms of its columns. ∎

2.2.4. Subspace decomposition based on a singular value decomposition. We next address the question of subspace decomposition; i.e., given $\hat{\mathcal{V}}_{j+1} = \begin{bmatrix}\mathcal{V}_j & \begin{bmatrix}P_{j-1} & \hat{V}_{j+1}\end{bmatrix}\end{bmatrix}$ obtained after the jth iteration of the deflated block Arnoldi procedure, we want to determine k_{j+1}, d_{j+1}, and the unitary matrix $\mathcal{F}_{j+1} \in \mathbb{C}^{(s_j+p)\times(s_j+p)}$ such that the decomposition (2.7) holds. To limit the computational cost related to the construction of $\mathcal{V}_{j+1}$, we consider the splitting $\mathcal{V}_{j+1} = \begin{bmatrix}\mathcal{V}_j & V_{j+1}\end{bmatrix}$ with $\mathcal{V}_j \in \mathbb{C}^{n\times s_j}$ obtained at the previous iteration and $V_{j+1} \in \mathbb{C}^{n\times k_{j+1}}$ to be determined. Thus the decomposition (2.7) can be written as

$$\begin{bmatrix}\mathcal{V}_j & \begin{bmatrix}V_{j+1} & P_j\end{bmatrix}\end{bmatrix} = \begin{bmatrix}\mathcal{V}_j & \begin{bmatrix}P_{j-1} & \hat{V}_{j+1}\end{bmatrix}\end{bmatrix} \mathcal{F}_{j+1}, \tag{2.13}$$

with $P_j \in \mathbb{C}^{n\times d_{j+1}}$ and $k_{j+1} + d_{j+1} = p$. Given the block form for $\mathcal{F}_{j+1}$,

$$\mathcal{F}_{j+1} = \begin{bmatrix}F_{11} & F_{12} \\ F_{21} & F_{22}\end{bmatrix},$$

where $F_{11} \in \mathbb{C}^{s_j \times s_j}$, $F_{12} \in \mathbb{C}^{s_j \times p}$, $F_{21} \in \mathbb{C}^{p\times s_j}$, and $F_{22} \in \mathbb{C}^{p\times p}$, the relation (2.13) becomes

$$\begin{bmatrix}\mathcal{V}_j & \begin{bmatrix}V_{j+1} & P_j\end{bmatrix}\end{bmatrix} = \begin{bmatrix}\mathcal{V}_j F_{11} + \begin{bmatrix}P_{j-1} & \hat{V}_{j+1}\end{bmatrix} F_{21} & \mathcal{V}_j F_{12} + \begin{bmatrix}P_{j-1} & \hat{V}_{j+1}\end{bmatrix} F_{22}\end{bmatrix}.$$

Since $\mathcal{V}_j^H \begin{bmatrix} P_{j-1} & \hat{V}_{j+1} \end{bmatrix} = 0_{s_j \times p}$ we deduce the following matrix structure:

$$\mathcal{F}_{j+1} = \begin{bmatrix} I_{s_j} & 0_{s_j \times p} \\ 0_{p \times s_j} & F_j \end{bmatrix}, \tag{2.14}$$

where the unitary matrix $F_j \in \mathbb{C}^{p \times p}$ remains to be determined. The criterion proposed in [39] to deduce F_j, k_{j+1}, and d_{j+1} aims at finding a possible linear combination of the columns of $R_j D^{-1}$ that are approximately dependent (with respect to a certain threshold) to determine the set of directions that we do not want to consider when defining V_{j+1} in $\mathcal{V}_{j+1} = \begin{bmatrix} \mathcal{V}_j & V_{j+1} \end{bmatrix}$. Since $R_j D^{-1} = \hat{\mathcal{V}}_{j+1} \hat{\mathcal{R}}_j$, we instead perform this analysis based on the singular value decomposition of $\hat{\mathcal{R}}_j$ as $\hat{\mathcal{R}}_j = U \Sigma W^H$. We note that the thin singular value decomposition of $\hat{\mathcal{R}}_j$ is rather inexpensive since $\hat{\mathcal{R}}_j$ does not depend on the problem size n. Heuristically, *tol* being the convergence threshold used in the stopping criterion of BGMRES-R, we first choose a relative positive deflation threshold ε_d and then determine k_{j+1} according to the following condition:

$$\sigma_l(\hat{\mathcal{R}}_j) > \varepsilon_d \; tol \quad \forall \; l \, \text{such that} \, 1 \leq l \leq k_{j+1}. \tag{2.15}$$

Since $d_{j+1} = p - k_{j+1}$, the following decomposition of $\hat{\mathcal{R}}_j$ at iteration j is then obtained with $\hat{\mathcal{R}}_{s_j} \in \mathbb{C}^{s_j \times p}$ and $\hat{\mathcal{R}}_p \in \mathbb{C}^{p \times p}$:

$$\hat{\mathcal{R}}_j = \begin{bmatrix} \hat{\mathcal{R}}_{s_j} \\ \hat{\mathcal{R}}_p \end{bmatrix} = \begin{bmatrix} U_{s_j}^+ \\ U_p^+ \end{bmatrix} \Sigma_+ W_+^H + \begin{bmatrix} U_{s_j}^- \\ U_p^- \end{bmatrix} \Sigma_- W_-^H, \tag{2.16}$$

with $U_+ \in \mathbb{C}^{(s_j+p) \times k_{j+1}}$, $U_- \in \mathbb{C}^{(s_j+p) \times d_{j+1}}$, $\Sigma_+ \in \mathbb{C}^{k_{j+1} \times k_{j+1}}$, $\Sigma_- \in \mathbb{C}^{d_{j+1} \times d_{j+1}}$, $W_+ \in \mathbb{C}^{p \times k_{j+1}}$, and $W_- \in \mathbb{C}^{p \times d_{j+1}}$. Based on this splitting, Robbé and Sadkane have then proposed performing such a subspace decomposition at iteration j:

$$\text{range}((I_n - \mathcal{V}_j \mathcal{V}_j^H) R_j D^{-1}) = \text{range}(V_{j+1}) \oplus \text{range}(P_j),$$

where

$$\begin{aligned} \text{range}(V_{j+1}) &= \text{range}((I_n - \mathcal{V}_j \mathcal{V}_j^H) R_j D^{-1} W_+), \\ \text{range}(P_j) &= \text{range}((I_n - \mathcal{V}_j \mathcal{V}_j^H) R_j D^{-1} W_-), \end{aligned}$$

that is, the k_{j+1} directions associated with $(I_n - \mathcal{V}_j \mathcal{V}_j^H) R_j D^{-1} W_+$ (the kept ones) lie in V_{j+1}, while the d_{j+1} directions associated with $(I_n - \mathcal{V}_j \mathcal{V}_j^H) R_j D^{-1} W_-$ (the deflated ones, i.e., postponed and reintroduced later in next iterations if necessary) lie in P_j. Due to (2.16), this decomposition is also equivalent to

$$\begin{aligned} \text{range}(V_{j+1}) &= \text{range}\Big(\begin{bmatrix} P_{j-1} & \hat{V}_{j+1} \end{bmatrix} \begin{bmatrix} U_p^+ \end{bmatrix} \Sigma_+ \Big), \\ \text{range}(P_j) &= \text{range}\Big(\begin{bmatrix} P_{j-1} & \hat{V}_{j+1} \end{bmatrix} \begin{bmatrix} U_p^- \end{bmatrix} \Sigma_- \Big). \end{aligned}$$

Since $\begin{bmatrix} V_{j+1} & P_j \end{bmatrix} = \begin{bmatrix} P_{j-1} & \hat{V}_{j+1} \end{bmatrix} F_j$, the unitary matrix F_j is then simply obtained as the orthogonal factor of the QR decomposition of the $p \times p$ matrix $\begin{bmatrix} U_p^+ & U_p^- \end{bmatrix}$. This decomposition is summarized later in section 4, Algorithm 2.

3. Modified block flexible GMRES with deflation at each iteration. In this section we present a modified block GMRES method with deflation at each iteration, which allows variable preconditioning and truncation, two features of significant interest when targeting the solution of large-scale non-Hermitian linear systems with possibly many right-hand sides. We first briefly introduce the motivations for these novelties and then describe the main mathematical properties of the resulting method, named BFGMRES-S.[3]

3.1. Motivations. As discussed in section 2.2.2, BGMRES-R relies on the subspace decomposition $\mathcal{V}_1 = \hat{\mathcal{V}}_1$ (relation (2.8)). At the first iteration of each cycle, $k_1 = p$ directions are effectively considered in the block orthonormalization procedure, including preconditioning and matrix-vector product phases. In BGMRES-R the norm minimization property induces a nonincreasing behavior of the number of selected directions k_j in a given cycle, as shown later in Proposition 3.3. However, performing no deflation at restart ($k_1 = p, d_1 = 0$) leads to a nonmonotone behavior of k_j along cycles (see the top-right panel of Figure 5.1 for an illustration), which may induce a significant additional computational overhead if the method is often restarted. The situation with possibly multiple cycles is precisely of interest in real life applications since a moderate restart size m is usually selected to limit the memory requirements when large-scale problems are considered and/or when the number of right-hand sides p is large. To circumvent this difficulty, we propose to incorporate the subspace decomposition at the beginning of each cycle of the block Krylov subspace method, leading to

$$\begin{bmatrix}\mathcal{V}_1 & P_0\end{bmatrix} = \hat{\mathcal{V}}_1 \mathcal{F}_1, \tag{3.1}$$

with $k_1 + d_1 = p$, $\mathcal{V}_1 \in \mathbb{C}^{n\times k_1}$, $P_0 \in \mathbb{C}^{n\times d_1}$, $\mathcal{F}_1 \in \mathbb{C}^{p\times p}$ with $d_1 \neq 0$ in general. The purpose of this whole section is to analyze the properties of the resulting modified block flexible GMRES with deflation at each iteration. First, we will show in section 3.4 that performing this subspace decomposition at the beginning of each cycle will ensure a nonincreasing behavior for k_j, the number of selected directions along cycles, which is a desirable property. This is a major difference between BFGMRES-S and BGMRES-R. Second, it turns out that this modification allows us to easily incorporate truncation in the block Krylov subspace method, as shown later in section 3.6. This is particularly useful when the number of right-hand sides is large. Third, we extend the block Krylov subspace method to the case of variable preconditioning, a mandatory feature when, e.g., iterative methods are used as preconditioners, as investigated later in section 5. This last property is described next.

3.2. Flexible deflated Arnoldi relation. In a given cycle of the modified block Krylov subspace method, we assume that the preconditioning operation at iteration j ($1 \leq j \leq m$) can be represented as $Z_j = M_j^{-1} V_j$, where $Z_j \in \mathbb{C}^{n\times k_j}$, $V_j \in \mathbb{C}^{n\times k_j}$, and $M_j \in \mathbb{C}^{n\times n}$ is supposed to be nonsingular. In this setting, the block orthonormalization procedure then leads to the following relation:

$$A\mathcal{Z}_j = \hat{\mathcal{V}}_{j+1} \hat{\mathcal{H}}_j, \tag{3.2}$$

where $\mathcal{Z}_j \in \mathbb{C}^{n\times s_j}$ (see Algorithm 1 for further details). Equation (3.2)—later called the flexible deflated Arnoldi relation—can be stated as

$$A\mathcal{Z}_j = \begin{bmatrix}\mathcal{V}_{j+1} & P_j\end{bmatrix} \mathcal{H}_j,$$

[3]The suffix "S" is used to emphasize that the method is based on a subspace selection at each iteration, in both the standard and truncated cases.

where $\begin{bmatrix}\mathcal{V}_{j+1} & P_j\end{bmatrix}$ is defined as in (2.7) and $\mathcal{H}_j = \mathcal{F}_{j+1}^H \hat{\mathcal{H}}_j$. Based on this flexible deflated Arnoldi relation, the block Krylov subspace method will minimize $||B-AX||_F$ over the space $X_0 + \text{range}(\mathcal{Z}_j Y D)$ with $Y \in \mathbb{C}^{s_j \times p}$.

3.3. Representation of the scaled initial block residual. At iteration j of a given cycle of BFGMRES-S ($1 \leq j \leq m$), we recursively define the quantity $\hat{\Lambda}_j \in \mathbb{C}^{(s_j+p)\times p}$ as

$$\hat{\Lambda}_j = \begin{bmatrix} \mathcal{F}_j^H \hat{\Lambda}_{j-1} \\ 0_{k_j \times p} \end{bmatrix}. \tag{3.3}$$

In the next lemma we derive the representation of the scaled initial block residual $R_0 D^{-1}$ with respect to the $\hat{\mathcal{V}}_{j+1}$ basis.

LEMMA 3.1. *In the modified block flexible GMRES with deflation at each iteration (BFGMRES-S), the scaled initial block residual $R_0 D^{-1}$ can be expressed in the $\hat{\mathcal{V}}_{j+1}$ basis as*

$$R_0 D^{-1} = \hat{\mathcal{V}}_{j+1} \hat{\Lambda}_j, \tag{3.4}$$

with $\hat{\Lambda}_j$ defined as in (3.3).

Proof. We prove this lemma by induction. Let $\mathcal{A}_j$ denote the assumption $R_0 D^{-1} = \hat{\mathcal{V}}_{j+1} \hat{\Lambda}_j$ at index j. We note that $\mathcal{A}_0$ holds by construction (see relation (2.1)). We suppose that $\mathcal{A}_{j-1}$ is satisfied and want to prove that $\mathcal{A}_{j-1}$ implies $\mathcal{A}_j$. Due to (2.7) and the unitary character of $\mathcal{F}_j$, the quantity $\hat{\mathcal{V}}_j \hat{\Lambda}_{j-1}$ can be expressed as

$$\hat{\mathcal{V}}_j \hat{\Lambda}_{j-1} = \begin{bmatrix}\mathcal{V}_j & P_{j-1}\end{bmatrix} \mathcal{F}_j^H \hat{\Lambda}_{j-1},$$

which can be written as

$$\begin{aligned} \hat{\mathcal{V}}_j \hat{\Lambda}_{j-1} &= \begin{bmatrix}\mathcal{V}_j & P_{j-1} & \hat{V}_{j+1}\end{bmatrix} \begin{bmatrix} \mathcal{F}_j^H \hat{\Lambda}_{j-1} \\ 0_{k_j \times p} \end{bmatrix} \\ &= \hat{\mathcal{V}}_{j+1} \hat{\Lambda}_j, \end{aligned}$$

due to (2.5) and (3.3), respectively. Since $\hat{\mathcal{V}}_j \hat{\Lambda}_{j-1} = R_0 D^{-1}$, $\mathcal{A}_j$ is then satisfied. ∎

Due to the initial subspace decomposition (3.1), we remark that the representation of the scaled initial block residual in the $\hat{\mathcal{V}}_{j+1}$ basis in BFGMRES-S involves the matrices $\mathcal{F}_l$ ($1 \leq l \leq j$). In BGMRES-R this representation differs (compare relations (2.9) and (3.3), respectively).

3.4. Minimization property. We denote by $Y_j \in \mathbb{C}^{s_j \times p}$ the solution of the reduced minimization problem $\mathcal{P}_s$ considered in BFGMRES-S:

$$\mathcal{P}_s : Y_j = \underset{Y \in \mathbb{C}^{s_j \times p}}{\text{argmin}} ||\hat{\Lambda}_j - \hat{\mathcal{H}}_j Y||_F, \tag{3.5}$$

with $\hat{\mathcal{H}}_j$ and $\hat{\Lambda}_j$ defined in (3.2) and (3.3), respectively. We denote by $\hat{\mathcal{R}}_j \in \mathbb{C}^{(s_j+p)\times p}$ the block residual of the reduced least-squares problem $\mathcal{P}_s$, i.e., $\hat{\mathcal{R}}_j = \hat{\Lambda}_j - \hat{\mathcal{H}}_j Y_j$ ($1 \leq j \leq m$), and define $\hat{\mathcal{R}}_0 \in \mathbb{C}^{p\times p}$ as $\hat{\mathcal{R}}_0 = \hat{\Lambda}_0$. We analyze in Proposition 3.2 the norm minimization property occurring in BFGMRES-S.

PROPOSITION 3.2. *In the modified version of the block Krylov subspace method with deflation at each iteration (BFGMRES-S), solving the reduced minimization problem $\mathcal{P}_s$ of* (3.5) *amounts to minimizing the Frobenius norm of the block true residual $||B - AX||_F$ over the space $X_0 + \text{range}(\mathcal{Z}_j Y D)$ at iteration j ($1 \leq j \leq m$) of a given cycle, i.e.,*

$$\underset{Y \in \mathbb{C}^{s_j \times p}}{\text{argmin}} ||\hat{\Lambda}_j - \hat{\mathcal{H}}_j Y||_F = \underset{Y \in \mathbb{C}^{s_j \times p}}{\text{argmin}} ||R_0 D^{-1} - A\mathcal{Z}_j Y||_F \tag{3.6}$$

$$= \underset{Y \in \mathbb{C}^{s_j \times p}}{\text{argmin}} ||B - A(X_0 + \mathcal{Z}_j Y D)||_F, \tag{3.7}$$

with $\hat{\mathcal{H}}_j$ and $\hat{\Lambda}_j$ defined in (3.2) *and* (3.3) *respectively.*

Proof. The proof follows the same lines as that of Proposition 2.1, now using relation (3.4) and the flexible deflated Arnoldi relation (3.2). ∎

3.5. Behavior of the number of selected k_j directions along convergence. We prove the important property that the number of new directions to consider in BFGMRES-S enjoys a nonincreasing behavior along convergence, as stated in Proposition 3.3.

PROPOSITION 3.3. *Denote by $k_{j,c}$ the number of Krylov directions effectively considered as best directions to keep at the jth iteration of the cth cycle of BFGMRES-S ($1 \leq j \leq m$ and $c \geq 1$), and assume that $\mathcal{Z}_j$ is of full column rank at iteration j of cycle c. Then the following relations are satisfied:*

$$\forall c, \quad k_{j+1,c} \leq k_{j,c}, \tag{3.8}$$

$$\forall c, \quad k_{1,c+1} = k_{m+1,c}. \tag{3.9}$$

Proof. BFGMRES-S is based on a standard norm minimization procedure, as recalled in Proposition 3.2. Hence at iteration j of cycle c, $R_j D^{-1}$ can be expressed as

$$R_j D^{-1} = (I_n - \mathcal{W}_j \mathcal{W}_j^H) R_{j-1} D^{-1},$$

where $\mathcal{W}_j \in \mathbb{C}^{n \times s_j}$ denotes a matrix whose columns form an orthonormal basis of $\text{range}(A\mathcal{Z}_j)$; see, e.g., [17, section 3.1]. From [28, Theorem 3.3.16] we conclude that the singular values of the scaled block true residual are monotonically decreasing; i.e.,

$$\forall\, i \mid 1 \leq i \leq p, \quad \sigma_i(R_j D^{-1}) \leq \sigma_i(R_{j-1} D^{-1}). \tag{3.10}$$

As stated in section 2.2.4 (relation (2.15)), the determination of $k_{j+1,c}$ is directly related to the singular values of $R_j D^{-1}$ in the cycle c. Hence from the inequality (3.10) we immediately deduce the relation (3.8). Finally the equality (3.9) is just due to the initial subspace decomposition (3.1) performed at the beginning of the $(c+1)$th cycle in BFGMRES-S. ∎

We deduce from Proposition 3.3 that we ensure a monotonically nonincreasing behavior for the number of k_j selected directions *along convergence* (as depicted later in the bottom-left panel of Figure 5.1) in BFGMRES-S. This is a major difference from BGMRES-R, where a nonincreasing behavior of k_j is guaranteed only inside a cycle and not along cycles. Indeed the equality (3.9) is not satisfied in BGMRES-R due to the initial subspace decomposition (2.8). Hence BFGMRES-S is not equivalent to BGMRES-R if deflation at the beginning of a cycle occurs.

3.6. Incorporating truncation. We first detail the subspace selection in BFGMRES-S when truncation in operations is performed, and then discuss consequences for the convergence properties. Truncation in BFGMRES-S corresponds to imposing an upper bound on the number of directions that we keep in the set of active directions. This constraint is imposed both in the initial subspace decomposition ($k_1 \leq p_f$, where $1 \leq p_f \leq p$) and at each iteration of the current cycle ($k_{j+1} \leq p_f$, $1 \leq j \leq m$). This mainly aims to reduce the computational cost of the cycle. Truncation implies just a modified selection of k_{j+1} and d_{j+1}, whereas $\mathcal{F}_{j+1}$ is obtained similarly as in section 2.2.4. More precisely, using the notation of section 2.2.4, we first choose the relative deflation threshold ε_d and define $p_d \in \mathbb{N}$ according to

$$\sigma_l(\hat{\mathcal{R}}_j) > \varepsilon_d \; tol \quad \forall \, l \, \text{such that} \, 1 \leq l \leq p_d. \tag{3.11}$$

Truncation then consists of defining k_{j+1} as $k_{j+1} = \min(p_d, p_f)$ and setting d_{j+1} as $d_{j+1} = p - k_{j+1}$. When $p_d > p_f$ we note that the inequality $\sigma_l(\hat{\mathcal{R}}_j) \leq \varepsilon_d \; tol$ does not hold for $p_f < l \leq p_d$. Hence the combination of residuals that have not approximately converged are indeed deflated. As in the nontruncated case, the corresponding directions are kept and later introduced if needed. We remark that both Propositions 3.2 and 3.3 hold in the truncated case (see the bottom-right panel of Figure 5.1 for an illustration). We stress the fact that no directions are discarded; this is the major difference with BFGMREST(m), a flexible variant of BFGMRES(m) based on deflation and truncation performed at restart only [13, section 3.2.1 and Algorithm 4].[4] Nevertheless, due to truncation, BFGMRES-S may require more iterations to converge than does its nontruncated version. However, this drawback has to be weighed against the reduced computational cost of the iterations when $p_d > p_f$. The subspace selection based on truncation is summarized later in section 4, Algorithm 2. Finally, we remark that performing truncation along cycles is made possible only because of the initial subspace decomposition (3.1) realized at the beginning of each cycle in BFGMRES-S.

4. Algorithmic details, computational cost, and memory requirements. We next present the algorithmic details of the methods introduced so far in sections 2 and 3. We conclude this section by analyzing the computational cost and memory requirements of BFGMRES-S.

4.1. Deflated block Arnoldi. Algorithm 1 introduces the jth iteration of the deflated block Arnoldi procedure with block modified Gram–Schmidt, assuming that deflation has occurred at the previous iteration ($d_j \neq 0$). If not, this algorithm then reduces to the standard flexible block Arnoldi procedure that is described in, e.g., [13, Algorithm 1]. As in standard block Arnoldi, Algorithm 1 proceeds by orthonormalizing AZ_j against all the previous preconditioned Krylov directions, but additionally, orthonormalization against P_{j-1} is performed (lines 10 and 11 of Algorithm 1). The block modified Gram–Schmidt version is presented in Algorithm 1, but a version of block Arnoldi due to Ruhe [40] or block Householder orthonormalization [3, 45] could be used as well.

4.2. Subspace decomposition. The subspace decomposition at the heart of the deflation at each iteration is described in Algorithm 2 and includes the possibility

[4] In addition, we note that BFGMRES-S can use truncation at each iteration, whereas BFGMREST(m) can use truncation only at the beginning of each cycle.

Algorithm 1. jth iteration of flexible deflated block Arnoldi with block modified Gram–Schmidt: Computation of $\hat{\mathcal{V}}_{j+1}$, $\mathcal{Z}_j$, and $s_j \in \mathbb{N}$ with $V_i \in \mathbb{C}^{n\times k_i}$ such that $V_i^H V_i = I_{k_i}$ $(1 \le i \le j)$, $p = k_j + d_j$, $P_{j-1} \in \mathbb{C}^{n\times d_j}$, and $[V_1, \ldots, V_j, P_{j-1}]^H [V_1, \ldots, V_j, P_{j-1}] = I_{s_{j-1}+p}$.

Define $s_{j-1} = \sum_{l=1}^{j-1} k_l$ $(s_0 = 0)$.
\# Choose preconditioning operator M_j^{-1}.
$Z_j = M_j^{-1} V_j$
$S = A Z_j$
\# Orthogonalization of S with respect to $[V_1, \ldots, V_j, P_{j-1}]$
for $i = 1, \ldots, j$ **do**
  $H_{i,j} = V_i^H S$
  $S = S - V_i H_{i,j}$
end for
$H_p = P_{j-1}^H S$
$S = S - P_{j-1} H_p$
Define $H_j \in \mathbb{C}^{(s_{j-1}+p)\times k_j}$ as $H_j^T = [H_{1,j}, \ldots, H_{j,j}, H_p]^T$.
Compute the QR decomposition of S as $S = QT$, $Q \in \mathbb{C}^{n\times k_j}$, and $T \in \mathbb{C}^{k_j \times k_j}$.
Set $\hat{V}_{j+1} = Q$, $H_{j+1,j} = T$.
Define $s_j = s_{j-1} + k_j$.
Define $\mathcal{Z}_j \in \mathbb{C}^{n\times s_j}$ as $\mathcal{Z}_j = [Z_1, \ldots, Z_j]$, $\mathcal{V}_j \in \mathbb{C}^{n\times s_j}$ as $\mathcal{V}_j = [V_1, \ldots, V_j]$, and $\hat{\mathcal{V}}_{j+1} \in \mathbb{C}^{n\times(s_j+p)}$ as $\hat{\mathcal{V}}_{j+1} = \begin{bmatrix}\mathcal{V}_j & P_{j-1} & \hat{V}_{j+1}\end{bmatrix}$ such that $A Z_j = \hat{\mathcal{V}}_{j+1} \begin{bmatrix} H_j \\ H_{j+1,j}\end{bmatrix}$.

Algorithm 2. Determination of k_{j+1}, d_{j+1}, and $\mathcal{F}_{j+1}$ $(0 \le j \le m)$.

Choose a relative deflation threshold ε_d and the upper bound p_f $(1 \le p_f \le p)$.
Compute the SVD of $\hat{\mathcal{R}}_j$ as $\hat{\mathcal{R}}_j = U \Sigma W^H$ with $U \in \mathbb{C}^{(s_j+p)\times p}$, $\Sigma \in \mathbb{C}^{p\times p}$, and $W \in \mathbb{C}^{p\times p}$.
Select p_d singular values of $\hat{\mathcal{R}}_j$ such that $\sigma_l(\hat{\mathcal{R}}_j) > \varepsilon_d \, tol$ for all l such that $1 \le l \le p_d$.
Set $k_{j+1} = \min(p_d, p_f)$ and $d_{j+1} = p - k_{j+1}$.
Define $U_p \in \mathbb{C}^{p\times p}$ as $U_p = U(s_j + 1 : s_j + p, 1 : p)$.
Compute the QR decomposition of U_p as $U_p = F_j T_j$, with $F_j \in \mathbb{C}^{p\times p}$, $F_j^H F_j = I_p$.
Define $\mathcal{F}_{j+1} \in \mathbb{C}^{(s_j+p)\times(s_j+p)}$ as $\mathcal{F}_{j+1} = \begin{bmatrix} I_{s_j} & 0_{s_j\times p} \\ 0_{p\times s_j} & F_j\end{bmatrix}$.

of truncation. The deflation threshold ε_d is usually fixed and does not depend on the cycle. The nontruncated variant of the algorithm introduced in section 2.2.4 is simply recovered by setting $p_f = p$. In practice, we point out that only the $p \times p$ F_j matrix has to be stored in memory.

4.3. Algorithm of modified block flexible GMRES with deflation at each iteration. Algorithm 3 introduces the modified block flexible GMRES method with deflation at each iteration. This algorithm is later named BFGMRES-S(m, p_f), where m denotes the maximal number of iterations performed in a given cycle and p_f the upper bound on the number of directions to consider at iteration j of a given cycle when performing truncation $(1 \le p_f \le p)$. The nontruncated variant is sim-

Algorithm 3. BFGMRES-S(m, p_f).

1: Choose a convergence threshold tol, a relative deflation threshold ε_d, the size of the restart m, the maximum number of cycles $cycle_{\max}$, and maximal number of directions to keep p_f.
2: Choose an initial guess $X_0 \in \mathbb{C}^{n\times p}$.
3: Compute the initial block residual $R_0 = B - AX_0$.
4: Define the scaling diagonal matrix $D \in \mathbb{C}^{p\times p}$ as $D = \text{diag}(b_1, \ldots, b_p)$ with $b_l = ||B(:, l)||_2$ for l such that $1 \le l \le p$.
5: Set $s_0 = 0$.
6: **for** $cycle = 1, cycle_{\max}$ **do**
7: Compute the QR decomposition of $R_0 D^{-1}$ as $R_0 D^{-1} = \hat{\mathcal{V}}_1 \hat{\Lambda}_0$ with $\hat{\mathcal{V}}_1 \in \mathbb{C}^{n\times p}$ and $\hat{\Lambda}_0 \in \mathbb{C}^{p\times p}$.
8: Determine deflation unitary matrix $\mathcal{F}_1 \in \mathbb{C}^{p\times p}$ and k_1, d_1 such that $k_1 + d_1 = p$ (see Algorithm 2), and set $s_1 = k_1$.
9: Define $\begin{bmatrix}\mathcal{V}_1 & P_0\end{bmatrix} = \hat{\mathcal{V}}_1 \mathcal{F}_1$, with $\mathcal{V}_1 \in \mathbb{C}^{n\times s_1}$ ($P_0 \in \mathbb{C}^{n\times d_1}$) as the first s_1 (last d_1) columns of $\hat{\mathcal{V}}_1 \mathcal{F}_1$, and define $V_1 = \mathcal{V}_1$.
10: **for** $j = 1, m$ **do**
11: *Completion of $\hat{\mathcal{V}}_{j+1}$, $\mathcal{Z}_j$, and $\hat{\mathcal{H}}_j$:* Apply Algorithm 1 to obtain $\mathcal{Z}_j \in \mathbb{C}^{n\times s_j}$, $\hat{\mathcal{V}}_{j+1} \in \mathbb{C}^{n\times(s_j+p)}$, and $\hat{\mathcal{H}}_j \in \mathbb{C}^{(s_j+p)\times s_j}$ such that

$$A\mathcal{Z}_j = \hat{\mathcal{V}}_{j+1}\hat{\mathcal{H}}_j \quad \text{with} \quad \hat{\mathcal{V}}_{j+1} = \left[V_1, V_2, \ldots, V_j, P_{j-1}, \hat{V}_{j+1}\right].$$

12: Set $\hat{\Lambda}_j \in \mathbb{C}^{(s_j+p)\times p}$ as $\hat{\Lambda}_j = \begin{bmatrix}\mathcal{F}_j^H \hat{\Lambda}_{j-1} \\ 0_{k_j\times p}\end{bmatrix}$.
13: Solve the minimization problem $\mathcal{P}_s$: $Y_j = \text{argmin}_{Y\in\mathbb{C}^{s_j\times p}} ||\hat{\Lambda}_j - \hat{\mathcal{H}}_j Y||_F$.
14: Compute $\hat{\mathcal{R}}_j = \hat{\Lambda}_j - \hat{\mathcal{H}}_j Y_j$.
15: **if** $||\hat{\mathcal{R}}_j(:, l)||_2 \le tol$, $\forall\ l \mid 1 \le l \le p$, **then**
16: Compute $X_j = X_0 + \mathcal{Z}_j Y_j D$; stop;
17: **end if**
18: Determine deflation unitary matrix $\mathcal{F}_{j+1} \in \mathbb{C}^{(s_j+p)\times(s_j+p)}$ and k_{j+1}, d_{j+1} such that $k_{j+1} + d_{j+1} = p$ (see Algorithm 2).
19: Set $s_{j+1} = s_j + k_{j+1}$.
20: Define $\begin{bmatrix}\mathcal{V}_{j+1} & P_j\end{bmatrix} = \hat{\mathcal{V}}_{j+1}\mathcal{F}_{j+1}$, with $\mathcal{V}_{j+1} \in \mathbb{C}^{n\times s_{j+1}}$ (or $P_j \in \mathbb{C}^{n\times d_{j+1}}$) as the first s_{j+1} (or last d_{j+1}) columns of $\hat{\mathcal{V}}_{j+1}\mathcal{F}_{j+1}$.
21: Define $\mathcal{H}_j = \mathcal{F}_{j+1}^H \hat{\mathcal{H}}_j$, with $\mathcal{H}_j \in \mathbb{C}^{(s_j+p)\times s_j}$.
22: **end for**
23: $X_m = X_0 + \mathcal{Z}_m Y_m D$
24: $R_m = B - AX_m$
25: Set $R_0 = R_m$ and $X_0 = X_m$.
26: **end for**

ply recovered if $p_f = p$ is satisfied. In such a case, the algorithm is simply named BFGMRES-S(m).

A comparison of BFGMRES-R due to Robbé and Sadkane [39] (Algorithm 4, given in the appendix for convenience) with BFGMRES-S (Algorithm 3) reveals the three main differences discussed in section 3: the initial subspace decomposition (performed at lines 8 and 9), the modified representation of the reduced right-

hand side (line 12), and the resulting different minimization problem to be solved (line 13).

4.4. Computational cost and memory requirements. The question of the total computational cost of BFGMRES-S is now addressed. For that purpose we summarize in Table 4.1 the costs occurring during a given cycle of BFGMRES-S(m, p_f) (considering Algorithms 1, 2, and 3), excluding matrix-vector products and preconditioning operations which are problem-dependent. We have included the costs proportional to both the size of the original problem n and the number of right-hand sides p, assuming a QR-factorization based on modified Gram–Schmidt and a Golub–Reinsch SVD;[5] see, e.g, [23, section 5.4.5] and [27, Appendix C] for further details on operation counts. The total cost of a given cycle is then found to grow as $C_1np^2 + C_2p^3 + C_3np$, and we note that this cost is always nonincreasing along convergence due to Proposition 3.3.

Compared to BGMRES-R, additional operations are related to the computations of $\mathcal{F}_1$ and $\hat{\Lambda}_j$, operations that behave as p^3. The computation of $\begin{bmatrix}\mathcal{V}_{j+1} & P_j\end{bmatrix}$ is in practice the most expensive one in a given iteration of BFGMRES-S(m, p_f). Concerning the truncated variant, the computational cost of a cycle will be reduced only if $p_d > p_f$, since the upper bound on k_{j+1} will then be active. This situation occurs at the beginning of the convergence due to the nonincreasing behavior of the singular values of $\hat{\mathcal{R}}_j$ shown in Proposition 3.3.

TABLE 4.1
Computational cost of a cycle of BFGMRES-S(m, p_f) (Algorithm 3). This excludes the cost of matrix-vector operations and preconditioning operations.

Step	Computational cost
Computation of R_0D^{-1}	np
QR factorization of R_0D^{-1}	$2np^2 + np$
Computation of $\mathcal{F}_1$	$14p^3$
Computation of $\begin{bmatrix}\mathcal{V}_1 & P_0\end{bmatrix}$	$2np^2$
Block Arnoldi procedure[6]	C_j
Computation of $\hat{\Lambda}_j$	$2(s_{j-1} + p)^2p$
Computation of Y_j	$2s_j^3 + 3ps_j^2$
Computation of $\hat{\mathcal{R}}_j$	$(2s_j + 1)(s_j + p)p$
Computation of $\mathcal{F}_{j+1}$	$4s_jp^2 + 14p^3$
Computation of $\begin{bmatrix}\mathcal{V}_{j+1} & P_j\end{bmatrix}$	$2np^2$
Computation of $\mathcal{H}_j$	$2p^3$
Computation of X_m	$np + (2n + 1)s_mp$

Concerning storage proportional to the problem size n, BFGMRES-S(m, p_f) requires R_m, X_0, X_m, $\mathcal{V}_{m+1}$, and $\mathcal{Z}_m$ leading to a memory requirement of $2ns_m + 4np$ at the end of a given cycle. Since s_m varies from cycle to cycle, an upper bound of the memory requirement can be given as $n(2m + 1)p + 3np$ when p linear systems have to be considered at the beginning of a given cycle. We note that the storage is monotonically decreasing along convergence, a feature than can be, for instance, exploited if dynamic memory allocation is used.

[5] The Golub–Reinsch SVD decomposition $R = U\Sigma V^H$ with $R \in \mathbb{C}^{m\times n}$ requires $4mn^2 + 8n^3$ operations when only Σ and V have to be computed.

[6] Algorithm 1: The block Arnoldi method based on modified Gram–Schmidt requires $\sum_{j=1}^{m}\sum_{i=1}^{j}(4nk_ik_j + nk_j + 4nd_jk_j)$ operations (lines 6 to 11) plus $\sum_{j=1}^{m} 2nk_j^2$ operations for the QR decomposition of S (line 13). Thus $C_j = \sum_{j=1}^{m}(\sum_{i=1}^{j}(4nk_ik_j + nk_j + 4nd_jk_j) + 2nk_j^2)$.

5. Numerical experiments. We investigate the numerical behavior of block flexible Krylov subspace methods including deflation at each iteration on a challenging application in geophysics where the situation of multiple right-hand sides is common. The source terms correspond to Dirac sources in this example. Thus the block right-hand side $B \in \mathbb{C}^{n\times p}$ is extremely sparse (only one nonzero element per column), and the initial block residual corresponds to a full rank matrix. We compare both BFGMRES-R(m) and BFGMRES-S(m) with various preconditioned iterative methods based on flexible (block) GMRES(m) with a zero initial guess (X_0) and a moderate value of the restart parameter m. The iterative procedures are stopped when the following condition is satisfied:

$$\frac{||B(:,l) - AX(:,l)||_2}{||B(:,l)||_2} \leq tol \qquad \forall\ l = 1, \ldots, p.$$

A primary concern will be to evaluate whether BFGMRES-S(m) can be efficient when solving problems with multiple right-hand sides both in terms of preconditioner applications and total computational cost. Finally, the tolerance is set to $tol = 10^{-5}$ in the numerical experiments, and we fix the parameter ϵ_d of Algorithm 2 to 1.

5.1. Acoustic full waveform inversion. We focus on a specific application in geophysics related to the simulation of wave propagation phenomena on Earth [47]. Given a three-dimensional physical domain Ω_p, the propagation of a wave field in a heterogeneous medium can be modeled by the Helmholtz equation written in the frequency domain:

$$-\frac{\partial^2 u}{\partial x^2} - \frac{\partial^2 u}{\partial y^2} - \frac{\partial^2 u}{\partial z^2} - \frac{(2\pi f)^2}{c^2(x,y,z)} u = g_s(\mathbf{x}), \quad \mathbf{x} = (x,y,z) \in \Omega_p. \tag{5.1}$$

u represents the pressure field in the frequency domain, c the variable acoustic-wave velocity in ms^{-1}, and f the frequency in Hertz. The source term $g_s(\mathbf{x}) = \delta(\mathbf{x} - \mathbf{x_s})$ represents a harmonic point source located at (x_s, y_s, z_s). A popular approach—the perfectly matched layer formulation (PML) [8, 9]—has been used in order to obtain a satisfactory near-boundary solution, without many artificial reflections. As in [13], we consider a second-order finite difference discretization of the Helmholtz equation (5.1) on a uniform equidistant Cartesian grid of size $n_x \times n_y \times n_z$. The same stability condition (12 points per wavelength) relating f, the frequency, to h, the mesh grid size, and $c(x,y,z)$, the heterogeneous velocity field, has been considered ($12fh = \min_{(x,y,z)\in\Omega_h} c(x,y,z)$). In consequence, A is a sparse complex matrix which is non-Hermitian and nonsymmetric due to the PML formulation that leads to complex-valued variable coefficients in the partial differential equation [37, Appendix A]. The resulting linear systems are known to be challenging for iterative methods [19, 20]. We consider the same approximate geometric two-level preconditioner presented in [13], which has been shown to be relatively efficient for the solution of three-dimensional heterogeneous Helmholtz problems in geophysics. We refer the reader to [13, Algorithm 5] for a complete description of the geometric preconditioner, and to [37] for additional theoretical properties in relation to Krylov subspace methods. In this section we consider this variable two-grid preconditioner in the multiple right-hand-side case and next investigate the performance of the block flexible Krylov methods on this challenging real-life application. The numerical results have been obtained on Babel, a Blue Gene/P computer located at IDRIS (PowerPC 450, 850 MHz, with 512 MB of memory on each core), using a Fortran 90 implementation with MPI

in single precision arithmetic. This code was compiled by the IBM compiler suite with standard compiling options and linked with the vendor BLAS and LAPACK subroutines.

As in [13], we consider the velocity field issued from the public domain SEG/EAGE Overthrust model [2] and analyze the performance of the numerical methods at a given frequency $f = 3.64$ Hz. Both the problem dimension (about 23 million unknowns) and the maximal number of right-hand sides to be considered (128) correspond to a task that geophysicists typically must face on a daily basis. Thus efficient numerical methods must be developed for that purpose. In [13] we have considered block flexible Krylov subspace methods including deflation at restart only for this application with a reduced number of right-hand sides (from 4 to 16). We continue this detailed analysis and investigate the performance of both BFGMRES-S(m, p_f) and BFGMRES-R(m) with a larger number of right-hand sides. In addition, we consider the standard block flexible GMRES method (BFGMRES(m)), the block flexible GMRES(m) with deflation performed at restart only (BFGMRESD(m) [13, Algorithm 3]), and the block flexible GMRES(m) with deflation and truncation performed at restart only (BFGMREST(m, p_f) [13, Algorithm 4]). We also investigate a combination of BFGMRES-S and BFGMRESD. This method, later named Combined(m, p_s), corresponds to BFGMRES-S(m) at the beginning of the convergence history. Then as soon as the number of Krylov directions effectively considered at iteration j (k_j) reaches a given prescribed value (p_s), the method switches to BFGMRESD(m) at the next restart. This mainly aims at reducing the computational cost in the next cycles by performing deflation *only* at the restart instead of at each iteration. Finally the number of cores is set to $8p$, ranging from 32 for $p = 4$ to 1024 for $p = 128$. This aims at imposing the same memory constraint on each core for all numerical experiments, as in [13]. The maximal memory requested is about 488 Gb for $p = 128$.

Table 5.1 collects, in addition to iterations (It)[7] and preconditioner applications on a single vector (Pr),[8] the computational times in seconds (T). Among the different strategies, BFGMRES-S(5) most often delivers the minimal number of preconditioner applications and computational times (see italic and bold values, respectively, in Table 5.1). This clearly highlights the value of performing deflation at each iteration, both in terms of preconditioner applications and computational operations on this given application. The improvement over BFGMRES-R(5) ranges from 10% for $p = 4$ to 35% for $p = 128$, which is very satisfactory behavior. BFGMRES-S(5) is also found to be competitive with respect to methods incorporating deflation at restart only (a gain of up to 15% in terms of computational time is obtained, for instance, for $p = 8$) as well as BFGMRES-S(5,p/2) (maximal gain of 21% (for $p = 32$) when compared to BFGMREST(5,p/2)). This is a satisfactory improvement, since methods including deflation at restart only are already quite efficient in this application, as shown in [13]. We also note that the improvement over the classical block flexible GMRES method is quite large as expected (a maximal gain of about 60% is obtained for $p = 64$).

We have also considered the solution of the p linear systems given now in sequence with the FGMRES Krylov subspace method [41]. In Table 5.1, FGMRES($5p$) consists of solving the p linear systems in sequence (starting with a zero initial guess),

[7] A complete cycle of BFGMRES(m), BFGMRES-R(m), or BFGMRES-S(m) always corresponds to m iterations, whereas a complete cycle of FGMRES(mp) involves mp iterations.

[8] A complete cycle of BFGMRES(m) corresponds to mp preconditioner applications, whereas a complete cycle of either BFGMRES-R(m) or BFGMRES-S(m) corresponds to $\sum_{j=1}^{m} k_{j,c}$ preconditioner applications. A complete cycle of FGMRES(mp) requires mp preconditioner applications.

TABLE 5.1

Acoustic full waveform inversion (SEG/EAGE Overthrust model). Case of $f = 3.64\ Hz$ ($h = 50\ m$), with $p = 4$ to $p = 128$ right-hand sides given at once. It denotes the number of iterations, Pr the number of preconditioner applications on a single vector, and T the total computational time in seconds. The number of cores is set to $8p$.

Acoustic full waveform inversion - *Grid* : $433 \times 433 \times 126$									
	$p = 4$			$p = 8$			$p = 16$		
Method	It	Pr	T	It	Pr	T	It	Pr	T
FGMRES($5p$)	56	56	624	112	112	629	224	224	665
BFGMRES(5)	14	56	622	14	112	631	14	224	668
BFGMRESD(5)	14	43	489	15	70	401	15	120	371
BFGMRES-R(5)	16	44	503	16	74	431	16	134	417
BFGMRES-S(5)	16	*39*	**452**	16	*57*	**339**	18	*102*	328
BFGMREST(5,p/2)	24	48	542	23	80	447	20	140	410
BFGMRES-S(5,p/2)	16	40	459	15	68	392	17	124	384
Combined(5,p/2)	15	41	471	15	62	359	15	103	323
Combined(5,p/4)	18	41	474	15	59	346	15	*102*	**320**
	$p = 32$			$p = 64$			$p = 128$		
Method	It	Pr	T	It	Pr	T	It	Pr	T
FGMRES($5p$)	434	434	670	1152	1152	925	2531	2531	1187
BFGMRES(5)	14	448	713	18	1152	962	19	2432	1187
BFGMRESD(5)	15	225	371	20	490	422	25	1015	509
BFGMRES-R(5)	18	283	466	25	618	537	28	1489	762
BFGMRES-S(5)	19	*181*	316	25	413	375	28	915	497
BFGMREST(5,p/2)	20	255	396	25	550	444	28	1125	524
BFGMRES-S(5,p/2)	16	189	310	24	444	396	29	976	523
Combined(5,p/2)	15	184	**305**	20	409	348	25	899	**442**
Combined(5,p/4)	20	191	320	20	*398*	**342**	25	*898*	448

the Euclidean norm of each residual being minimized over a subspace of maximal dimension $5p$. The maximal number of iterations performed to reach the stopping criterion (5.1) on a single linear system is found to be equal to 14 (p ranging from 4 to 32), 18 ($p = 64$), and 22 ($p = 128$), respectively. These results lead to two important comments. First, whatever the number of right-hand sides considered, no restart occurs in the Krylov subspace method applied in a single right-hand side situation: FGMRES($5p$) thus corresponds to a preconditioned full flexible GMRES method in such a case. This is thus ideal for FGMRES($5p$), since no restart procedure that might have hampered the convergence of the method is involved. Second, we remark that the maximal number of iterations performed does depend on the number of cores. This behavior can be explained as follows. An analysis of the FGMRES Krylov subspace method with the variable two-grid preconditioner on three-dimensional heterogeneous Helmholtz problems has shown that the numerical method satisfies a strong scalability property up to a given number of cores [37]. The loss of scalability is indeed due to the symmetric Gauss-Seidel preconditioner used both in the smoother and in the approximate solution of the coarse problem. This preconditioner is based on a subdomain decoupling and thus becomes inherently less efficient when the number of cores is increasing [6]. We refer the reader to [37] and [13, section 4.2.2] for related numerical experiments and additional comments. Finally, we remark that the improvement due to block methods using deflation at each iteration over the flexible GMRES method applied on the sequence of linear systems is noticeable on this application; a maximal gain of about 62% is obtained for $p = 128$.

TABLE 5.2

Acoustic full waveform inversion (SEG/EAGE Overthrust model). Case of $f = 3.64\ Hz$ ($h = 50\ m$), with $p = 4$ to $p = 128$ right-hand sides given at once. Detailed timings (in seconds) related to orthogonalization (T_{orth}) and to preconditioning and matrix-vector products (T_{pmvp}). Here $\sigma = T_{pmvp}/T$ represents the percentage of time spent in the preconditioning and matrix-vector product phases with respect to the total computational times (T) given in Table 5.1. The number of cores is set to $8p$.

Acoustic full waveform inversion - *Grid* : $433 \times 433 \times 126$									
	$p = 4$			$p = 8$			$p = 16$		
Method	T_{orth}	T_{pmvp}	σ	T_{orth}	T_{pmvp}	σ	T_{orth}	T_{pmvp}	σ
FGMRES($5p$)	10	607	0.97	8	609	0.97	5	646	0.97
BFGMRES(5)	13	605	0.97	17	608	0.96	29	631	0.94
BFGMRESD(5)	14	470	0.96	11	384	0.96	16	348	0.94
BFGMRES-R(5)	15	480	0.95	14	408	0.95	18	386	0.93
BFGMRES-S(5)	16	428	0.95	12	317	0.94	15	299	0.91
BFGMREST(5,p/2)	16	519	0.96	10	425	0.95	11	391	0.95
BFGMRES-S(5,p/2)	15	436	0.95	10	373	0.95	15	357	0.93
Combined(5,p/2)	16	449	0.95	10	343	0.96	14	301	0.93
Combined(5,p/4)	16	449	0.95	12	328	0.95	13	298	0.93
	$p = 32$			$p = 64$			$p = 128$		
Method	T_{orth}	T_{pmvp}	σ	T_{orth}	T_{pmvp}	σ	T_{orth}	T_{pmvp}	σ
FGMRES($5p$)	9	614	0.92	15	862	0.93	25	1141	0.96
BFGMRES(5)	51	649	0.91	116	818	0.85	223	906	0.76
BFGMRESD(5)	25	334	0.90	45	354	0.84	74	385	0.76
BFGMRES-R(5)	27	417	0.89	30	428	0.80	98	565	0.74
BFGMRES-S(5)	19	275	0.87	31	300	0.80	56	348	0.70
BFGMREST(5,p/2)	16	368	0.93	29	389	0.88	50	423	0.81
BFGMRES-S(5,p/2)	17	276	0.89	31	320	0.81	58	371	0.71
Combined(5,p/2)	17	276	0.90	28	297	0.85	50	342	0.77
Combined(5,p/4)	18	286	0.89	27	288	0.84	51	341	0.76

Detailed computational timings spent in the orthogonalization phase (T_{orth}) and in both preconditioning and outer matrix-vector product phases (T_{pmvp}) are provided in Table 5.2. In addition the percentages (σ) of time spent in the preconditioning and matrix-vector product phases with respect to the total computational times are given. The analysis of σ clearly highlights that the dominant cost in all the methods is related to the preconditioning phase, which is in agreement with the main assumption of the paper. In the application, the approximate solution of the coarse linear system obtained with a symmetric Gauss–Seidel preconditioned restarted GMRES method represents the most computationally expensive part of the two-grid cycle used as a preconditioner. We refer the reader to [37] for further details on the preconditioner.

Figure 5.1 shows the evolution of k_j along convergence for the various block subspace methods in the case of $p = 32$. Regarding BFGMRESD(5) and BFGMREST(5,p/2) deflation is performed only at the beginning of each cycle; thus k_j is found to be constant in a given cycle. Variations at each iteration can happen only in BFGMRES-R(5) or in BFGMRES-S(5). As expected, BFGMRES-S(5) enjoys a nonincreasing behavior for k_j along convergence, while peaks occur for BFGMRES-R(5) at the beginning of each cycle (see Proposition 3.3). In this example the use of truncation within BFGMRES-S(5, p/2) tends to delay the beginning of the decreasing behavior of k_j. After a certain phase deflation is nevertheless active and proves to be useful.

We also remark that the use of truncation techniques in BFGMRES-S(m, p_f) leads to an efficient method. In certain cases BFGMRES-S(5, p/2) is as efficient as

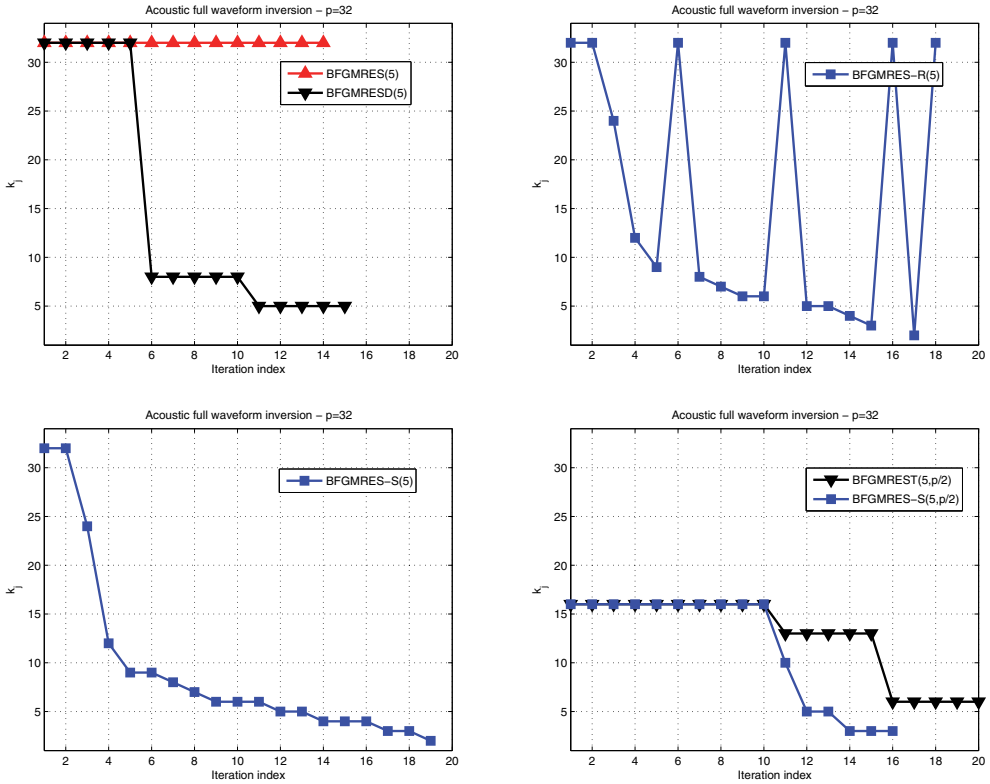

FIG. 5.1. *Acoustic full waveform inversion (SEG/EAGE Overthrust model). Case of* $p = 32$. *Evolution of* k_j *versus iterations for* $p = 32$ *in BFGMRES(5) and BFGMRESD(5) (top-left), BFGMRES-R(5) (top-right), BFGMRES-S(5) (bottom-left), and truncated variants (BFGMREST(5,p/2) and BFGMRES-S(5,p/2)) (bottom-right).*

BFGMRES-S(5) in terms of computational times (see, e.g., the case $p = 32$ in Table 5.1). This feature is really important in the given application due to the large size of the linear systems. Furthermore BFGMRES-S(5, p/2) usually requires fewer preconditioner applications than does BFGMREST(5, p/2). This satisfactory behavior has a definite reason: due to Proposition 3.2, we guarantee that the truncated variant of BFGMRES-S(m, p_f) minimizes the whole residual at each iteration (regardless of the value of p_f), whereas BFGMREST(m) chooses just a set of linear independent columns of the block residual to be minimized at each cycle. We consider that this is indeed a critical feature of the truncated variant of BFGMRES-S(m, p_f). Furthermore, as shown in Table 5.1, the Combined$(5, p_s)$ method (with $p_s = p/2$ or $p_s = p/4$) leads to further reductions in computational times and is especially appropriate when the number of right-hand sides becomes large on this given application.

Finally, in [12, section 6.1] and [30, section 3.9.3] the first five strategies (FGMRES(mp), BFGMRES(m), BFGMRESD(m), BFGMRES-R(m), and BFGMRES-S(m)) have been evaluated on an academic test case related to a two-dimensional partial differential equation (complex-valued advection-diffusion reaction problem) with a number of right-hand sides ranging from 4 to 32. A cycle of GMRES(m) has been used as a variable preconditioner in all methods. Whatever the value of the restart

parameter m (two values have been considered, $m = 5$ and $m = 10$), it was found that BFGMRES-S(m) always led to the minimal number of preconditioner applications and delivered the best efficiency in terms of computational operations. This is thus behavior similar to the proposed application in geophysics. We also refer the reader to [30, section 3.9] for additional numerical experiments on academic problems related to partial differential equations showing a similar trend.

6. Conclusion. We have proposed a block restarted GMRES method for the solution of non-Hermitian linear systems with multiple right-hand sides that allows both the use of deflation at each iteration and variable preconditioning. This method uses a subspace decomposition based on the singular value decomposition of the block residual of the reduced least-squares problem. This decomposition aims at selecting a set of k_j new Krylov directions at iteration j, while d_j directions are deflated (i.e., kept and reintroduced later if needed) at the same iteration. The new method ensures a nonincreasing behavior of k_j along convergence, which leads to possibly considerable computational savings with respect to the existing reference method [39]. We have also proposed a variant based on truncation. All these features are particularly of interest when tackling the solution of large-scale linear systems with many right-hand sides. BFGMRES-S has proved to be efficient in terms of both preconditioner applications and computational operations on an application related to geophysics. Often, *but not always*, it has been found superior to recent block flexible methods including deflation at restart only. We would like to emphasize that, when large restart sizes m or large numbers of right-hand sides p are considered, the cost of orthogonalization can become significant. In consequence this may potentially decrease the value of performing deflation at each iteration. Nevertheless, in this paper, satisfactory behavior has been observed on an industrial simulation, where large linear systems with multiple right-hand sides have been successfully solved in a parallel distributed memory environment. Further reductions in terms of computational times have been obtained by combining methods including deflation at each iteration and deflation at restart only in a second phase.

It is worth noting that the theoretical properties of BFGMRES-S hold for any unitary matrix $\mathcal{F}_{j+1}$. Hence different subspace decompositions could be investigated. We also note that the analysis proposed in this paper can be extended as well to other block Krylov subspace methods based on a norm minimization property, such as block FOM [38], block GCRO [49], and block simpler GMRES [32]. All these methods do rely on block orthogonalizations that require global communications. These latter operations usually become a bottleneck on massively parallel platforms, and we plan in the near future to investigate algorithmic variants, where these global communications can be overlapped with calculations or local communications. This is especially interesting for large-scale problems.

To give a broader picture of the performance of the block Krylov subspace methods investigated here, we finally mention that a comparison with flexible variants of block Lanczos algorithms including deflation at each iteration should be performed. This is the topic of a forthcoming study.

Appendix. Algorithm 4 shows the restarted block GMRES method with deflation at each iteration in the case of variable preconditioning that is considered in section 5. This algorithm is named BFGMRES-R(m). We note that the original algorithm [39, Algorithm 2] is simply recovered if each preconditioning operator M_j is chosen as the identity operator I_n in Algorithm 1.

Algorithm 4. BFGMRES-R(m) [39].

1: Choose a convergence threshold tol, a relative deflation threshold ε_d, the size of the restart m, and the maximum number of cycles $cycle_{\max}$.
2: Choose an initial guess $X_0 \in \mathbb{C}^{n\times p}$.
3: Compute the initial block residual $R_0 = B - AX_0$.
4: Define the scaling diagonal matrix $D \in \mathbb{C}^{p\times p}$ as $D = \mathrm{diag}(b_1, \ldots, b_p)$ with $b_l = ||B(:,l)||_2$ for l such that $1 \le l \le p$.
5: Set $s_0 = 0$.
6: **for** $cycle = 1, cycle_{\max}$ **do**
7: Compute the QR decomposition of $R_0 D^{-1}$ as $R_0 D^{-1} = \hat{\mathcal{V}}_1 \hat{\Lambda}_0$ with $\hat{\mathcal{V}}_1 \in \mathbb{C}^{n\times p}$ and $\hat{\Lambda}_0 \in \mathbb{C}^{p\times p}$.
8: Set $k_1 = p$, $d_1 = 0$, and $s_1 = k_1$.
9: Define[9] $\begin{bmatrix}\mathcal{V}_1 & P_0\end{bmatrix} = \hat{\mathcal{V}}_1$, with $\mathcal{V}_1 \in \mathbb{C}^{n\times s_1}$ ($P_0 \in \mathbb{C}^{n\times d_1}$) as the first s_1 (last d_1) columns of $\hat{\mathcal{V}}_1$, and define $V_1 = \mathcal{V}_1$.
10: **for** $j = 1, m$ **do**
11: *Completion of $\hat{\mathcal{V}}_{j+1}$, $\mathcal{Z}_j$, and $\hat{\mathcal{H}}_j$:* Apply Algorithm 1 to obtain $\mathcal{Z}_j \in \mathbb{C}^{n\times s_j}$, $\hat{\mathcal{V}}_{j+1} \in \mathbb{C}^{n\times(s_j+p)}$, and $\hat{\mathcal{H}}_j \in \mathbb{C}^{(s_j+p)\times s_j}$ such that

$$A\mathcal{Z}_j = \hat{\mathcal{V}}_{j+1}\hat{\mathcal{H}}_j \quad \text{with} \quad \hat{\mathcal{V}}_{j+1} = \left[V_1, V_2, \ldots, V_j, P_{j-1}, \hat{V}_{j+1}\right].$$

12: Set $\hat{\Lambda}_j \in \mathbb{C}^{(s_j+p)\times p}$ as $\hat{\Lambda}_j = \begin{bmatrix}\hat{\Lambda}_0 \\ 0_{s_j\times p}\end{bmatrix}$.
13: Solve the minimization problem $\mathcal{P}_r$: $Y_j = \mathrm{argmin}_{Y\in\mathbb{C}^{s_j\times p}} ||\hat{\Lambda}_j - \hat{\mathcal{H}}_j Y||_F$.
14: Compute $\hat{\mathcal{R}}_j = \hat{\Lambda}_j - \hat{\mathcal{H}}_j Y_j$.
15: **if** $||\hat{\mathcal{R}}_j(:,l)||_2 \le tol \ \forall\ l \mid 1 \le l \le p$, **then**
16: Compute $X_j = X_0 + \mathcal{Z}_j Y_j D$; stop;
17: **end if**
18: Determine deflation unitary matrix $\mathcal{F}_{j+1} \in \mathbb{C}^{(s_j+p)\times(s_j+p)}$ and k_{j+1}, d_{j+1} such that $k_{j+1} + d_{j+1} = p$ (see Algorithm 2 with $p_f = p$).
19: Set $s_{j+1} = s_j + k_{j+1}$.
20: Define $\begin{bmatrix}\mathcal{V}_{j+1} & P_j\end{bmatrix} = \hat{\mathcal{V}}_{j+1}\mathcal{F}_{j+1}$, with $\mathcal{V}_{j+1} \in \mathbb{C}^{n\times s_{j+1}}$ (or $P_j \in \mathbb{C}^{n\times d_{j+1}}$) as the first s_{j+1} (or last d_{j+1}) columns of $\hat{\mathcal{V}}_{j+1}\mathcal{F}_{j+1}$.
21: Define $\mathcal{H}_j = \mathcal{F}_{j+1}^H \hat{\mathcal{H}}_j$, with $\mathcal{H}_j \in \mathbb{C}^{(s_j+p)\times s_j}$.
22: **end for**
23: $X_m = X_0 + \mathcal{Z}_m Y_m D$
24: $R_m = B - AX_m$
25: Set $R_0 = R_m$ and $X_0 = X_m$.
26: **end for**

Acknowledgments. The authors would like to thank Michele Benzi and the two referees for their comments and suggestions, which considerably helped to improve the manuscript. The authors would like to acknowledge GENCI (Grand Equipement National de Calcul Intensif) for the donation of computing hours on the IBM Blue Gene/P computer at IDRIS, France. The authors were granted access to the HPC resources of IDRIS under allocation 2012065068 and 2013065068 made by GENCI.

[9] We have made the abuse of notation $\begin{bmatrix}\mathcal{V}_1 & P_0\end{bmatrix} = \hat{\mathcal{V}}_1$ to allow an easy-to-read comparison with line 9 of Algorithm 3. In BFGMRES-R(m) we have $\mathcal{V}_1 = \hat{\mathcal{V}}_1$ and $P_0 = [\,]$ in practice.

REFERENCES

[1] J. I. Aliaga, D. L. Boley, R. W. Freund, and V. Hernández, *A Lanczos-type method for multiple starting vectors*, Math. Comput., 69 (2000), pp. 1577–1601.

[2] F. Aminzadeh, J. Brac, and T. Kunz, *3D Salt and Overthrust Models*, Society of Exploration Geophysicists, Tulsa, OK, 1997.

[3] J. Baglama, *Augmented block Householder Arnoldi method*, Linear Algebra Appl., 429 (2008), pp. 2315–2334.

[4] Z. Bai, D. Day, and Q. Ye, *ABLE: An adaptive block Lanczos for non-Hermitian eigenvalue problems*, SIAM J. Matrix Anal. Appl., 20 (1999), pp. 1060–1082.

[5] A. H. Baker, J. M. Dennis, and E. R. Jessup, *On improving linear solver performance: A block variant of GMRES*, SIAM J. Sci. Comput., 27 (2006), pp. 1608–1626.

[6] A. H. Baker, R. D. Falgout, T. V. Kolev, and U. M. Yang, *Multigrid smoothers for ultraparallel computing*, SIAM J. Sci. Comput., 33 (2011), pp. 2864–2887.

[7] G. Barbella, F. Perotti, and V. Simoncini, *Block Krylov subspace methods for the computation of structural response to turbulent wind*, Comput. Methods Appl. Mech. Engrg., 200 (2011), pp. 2067–2082.

[8] J.-P. Berenger, *A perfectly matched layer for absorption of electromagnetic waves*, J. Comput. Phys., 114 (1994), pp. 185–200.

[9] J.-P. Berenger, *Three-dimensional perfectly matched layer for absorption of electromagnetic waves*, J. Comput. Phys., 127 (1996), pp. 363–379.

[10] W. E. Boyse and A. A. Seidl, *A block QMR method for computing multiple simultaneous solutions to complex symmetric systems*, SIAM J. Sci. Comput., 17 (1996), pp. 263–274.

[11] P. A. Businger and G. Golub, *Linear least squares solutions by Householder transformations*, Numer. Math., 7 (1965), pp. 269–276.

[12] H. Calandra, S. Gratton, R. Lago, and X. Vasseur, *A Deflated Minimal Block Residual Method for the Solution of non-Hermitian Linear Systems with Multiple Right-Hand Sides*, Technical Report TR/PA/12/45, CERFACS, Toulouse, France, 2012.

[13] H. Calandra, S. Gratton, J. Langou, X. Pinel, and X. Vasseur, *Flexible variants of block restarted GMRES methods with application to geophysics*, SIAM J. Sci. Comput., 34 (2012), pp. A714–A736.

[14] J. Cullum and T. Zhang, *Two-sided Arnoldi and nonsymmetric Lanczos algorithms*, SIAM J. Matrix Anal. Appl., 24 (2002), pp. 303–319.

[15] R. D. da Cunha and D. Becker, *Dynamic block GMRES: An iterative method for block linear systems*, Adv. Comput. Math., 27 (2007), pp. 423–448.

[16] L. Du, T. Sogabe, B. Yu, Y. Yamamoto, and S.-L. Zhang, *A block IDR(s) method for nonsymmetric linear systems with multiple right-hand sides*, J. Comput. Appl. Math., 235 (2011), pp. 4095–4106.

[17] M. Eiermann and O. Ernst, *Geometric aspects of the theory of Krylov subspace methods*, Acta Numer., 10 (2001), pp. 251–312.

[18] H. Elman, O. Ernst, D. O'Leary, and M. Stewart, *Efficient iterative algorithms for the stochastic finite element method with application to acoustic scattering*, Comput. Methods Appl. Mech. Engrg., 194 (2005), pp. 1037–1055.

[19] Y. A. Erlangga, *Advances in iterative methods and preconditioners for the Helmholtz equation*, Arch. Comput. Methods Engrg., 15 (2008), pp. 37–66.

[20] O. Ernst and M. J. Gander, *Why it is difficult to solve Helmholtz problems with classical iterative methods*, in Numerical Analysis of Multiscale Problems, O. Lakkis, I. Graham, T. Hou, and R. Scheichl, eds., Springer, New York, 2011, pp. 325–361.

[21] R. W. Freund, *Krylov-subspace methods for reduced-order modeling in circuit simulation*, J. Comput. Appl. Math., 123 (2000), pp. 395–421.

[22] R. W. Freund and M. Malhotra, *A block QMR algorithm for non-Hermitian linear systems with multiple right-hand sides*, Linear Algebra Appl., 254 (1997), pp. 119–157.

[23] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed., Johns Hopkins University Press, Baltimore, 1996.

[24] A. El Guennouni, K. Jbilou, and H. Sadok, *A block version of BICGSTAB for linear systems with multiple right-hand sides*, Electron. Trans. Numer. Anal., 16 (2003), pp. 129–142.

[25] 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, A. H. Siddiqi, I. S. Duff, and O. Christensen, eds., Anamaya Publishers, New Delhi, India, 2006, pp. 420–447.

[26] M. H. Gutknecht and T. Schmelzer, *The block grade of a block Krylov space*, Linear Algebra Appl., 430 (2009), pp. 174–185.

[27] N. J. Higham, *Functions of Matrices: Theory and Computation*, SIAM, Philadelphia, 2008.
[28] R. Horn and C. R. Johnson, *Topics in Matrix Analysis*, Cambridge University Press, Cambridge, UK, 1991.
[29] A. Khabou, *Solveur itératif haute performance pour les systèmes linéaires avec seconds membres multiples*, Master's thesis, Department of Computer Science, University of Bordeaux I, 2009.
[30] R. Lago, *A Study on Block Flexible Iterative Solvers with Applications to Earth Imaging Problem in Geophysics*, Ph.D. thesis, CERFACS, Toulouse, France, 2013.
[31] J. Langou, *Iterative methods for solving linear systems with multiple right-hand sides*, Ph.D. thesis, Department of Mathematics, CERFACS, Toulouse, France 2003.
[32] H. Liu and B. Zhong, *A simpler block GMRES for nonsymmetric systems with multiple right-hand sides*, Electron. Trans. Numer. Anal., 30 (2008), pp. 1–9.
[33] D. Loher, *Reliable Nonsymmetric Block Lanczos Algorithms*, Ph.D. thesis, Department of Mathematics, Swiss Federal Institute of Technology Zurich (ETHZ), Zurich, Switzerland, 2006.
[34] M. Malhotra, R. W. Freund, and P. M. Pinsky, *Iterative solution of multiple radiation and scattering problems in structural acoustics using a block quasi-minimal residual algorithm*, Comput. Methods Appl. Mech. Engrg., 146 (1997), pp. 173–196.
[35] A. A. Nikishin and A. Yu. Yeremin, *Variable block CG algorithms for solving large sparse symmetric positive definite linear systems on parallel computers,* I: *General iterative scheme*, SIAM J. Matrix Anal. Appl., 16 (1995), pp. 1135–1153.
[36] D. P. O'Leary, *The block conjugate gradient algorithm and related methods*, Linear Algebra Appl., 29 (1980), pp. 293–322.
[37] X. Pinel, *A Perturbed Two-level Preconditioner for the Solution of Three-dimensional Heterogeneous Helmholtz Problems with Applications to Geophysics*, Ph.D. thesis, Department of Mathematics, CERFACS, Toulouse, France, 2010.
[38] M. Robbé and M. Sadkane, *Exact and Inexact Breakdowns in Block Versions of FOM and GMRES Methods*, Technical Report, Département de Mathématiques, Université de Bretagne Occidentale, Brest, France, 2004; available online at http://www.math.univ-brest.fr/archives/recherche/prepub/Archives/2005/breakdowns.pdf.
[39] M. Robbé and M. Sadkane, *Exact and inexact breakdowns in the block GMRES method*, Linear Algebra Appl., 419 (2006), pp. 265–285.
[40] A. Ruhe, *Implementation aspects of band Lanczos algorithms for computation of eigenvalues of large sparse symmetric matrices*, Math. Comput., 33 (1979), pp. 680–687.
[41] Y. Saad, *A flexible inner-outer preconditioned GMRES algorithm*, SIAM J. Sci. Comput., 14 (1993), pp. 461–469.
[42] Y. Saad, *Iterative Methods for Sparse Linear Systems*, 2nd ed., SIAM, Philadelphia, 2003.
[43] T. Sakurai, H. Tadano, and Y. Kuramashi, *Application of block Krylov subspace algorithms to the Wilson-Dirac equation with multiple right-hand sides in lattice QCD*, Comput. Phys. Commun., 181 (2010), pp. 113–117.
[44] P. Soudais, *Iterative solution methods of a 3-D scattering problem from arbitrary shaped multidielectric and multiconducting bodies*, IEEE Trans. Antennas and Propagation, 42 (1994), pp. 954–959.
[45] X. Sun and C. Bischof, *A basis-kernel representation of orthogonal matrices*, SIAM J. Matrix Anal. Appl., 16 (1995), pp. 1184–1196.
[46] A. Toselli and O. Widlund, *Domain Decomposition Methods—Algorithms and Theory*, Springer Ser. Comput. Math. 34, Springer, New York, 2004.
[47] J. Virieux and S. Operto, *An overview of full waveform inversion in exploration geophysics*, Geophys., 74 (2009), pp. WCC127–WCC152.
[48] B. Vital, *Etude de quelques méthodes de résolution de problème linéaire de grande taille sur multiprocesseur*, Ph.D. thesis, Department of Computer Science, Université de Rennes, Rennes, France, 1990.
[49] R. Yu, E. de Sturler, and D. D. Johnson, *A Block Iterative Solver for Complex Non-Hermitian Systems Applied to Large-Scale Electronic-Structure Calculations*, Technical Report UIUCDCS-R-2002-2299, Department of Computer Science, University of Illinois at Urbana-Champaign, Champaign, IL, 2002.