IMA Journal of Numerical Analysis (2018) **00**, 1–33
doi:10.1093/imanum/dry054

Enlarged GMRES for solving linear systems with one or multiple right-hand sides

Hussam Al Daas* and Laura Grigori
Inria Paris, Alpines and UPMC Univ. Paris 06, CNRS UMR 7598, Laboratoire Jacques-Louis Lions
*Corresponding author: hussam.al-daas@inria.fr laura.grigori@inria.fr

Pascal Hénon
TOTAL EP, Centre Scientifique et Technique Jean Féger, Pau, France
pascal.henon@total.com

and

Philippe Ricoux
TOTAL SA, R&D Group, Tour Michelet La Défense 10, Paris-La Defense, France
philippe.ricoux@total.com

[Received on 15 March 2017; revised on 25 June 2018]

We propose a variant of the generalized minimal residual (GMRES) method for solving linear systems of equations with one or multiple right-hand sides. Our method is based on the idea of the enlarged Krylov subspace to reduce communication. It can be interpreted as a block GMRES method. Hence, we are interested in detecting inexact breakdowns. We introduce a strategy to perform the test of detection. Furthermore, we propose a technique for deflating eigenvalues that has two benefits. The first advantage is to avoid the plateau of convergence after the end of a cycle in the restarted version. The second is to have very fast convergence when solving the same system with different right-hand sides, each given at a different time (useful in the context of a constrained pressure residual preconditioner). We test our method with these deflation techniques on academic test matrices arising from solving linear elasticity and convection–diffusion problems as well as matrices arising from two real-life applications, seismic imaging and simulations of reservoirs. With the same memory cost we obtain a saving of up to 50% in the number of iterations required to reach convergence with respect to the original method.

Keywords: Krylov iterative methods; linear solvers; multiple right-hand sides; inexact breakdowns; deflation of eigenvalues; communication avoiding.

1. Introduction

In this paper $A \in \mathbb{C}^{n\times n}$ is a nonsingular non-Hermitian matrix. Let the system of linear equations be

$$AX = B, \tag{1.1}$$

where $X \in \mathbb{C}^{n\times s}$, and $B \in \mathbb{C}^{n\times s}$ is full rank, with $s \geqslant 1$ the number of right-hand sides. Here, we suppose that $s \ll n$. Block Krylov subspace methods are iterative schemes used to solve this type of linear system of equations. They find a sequence of approximate solutions $X_1, \ldots, X_j$ respectively in the

© The Author(s) 2018. Published by Oxford University Press on behalf of the Institute of Mathematics and its Applications. All rights reserved.

Downloaded from https://academic.oup.com/imajna/advance-article-abstract/doi/10.1093/imanum/dry054/5078636
by BIUS Jussieu user
on 28 August 2018

affine spaces $X_0 + \mathscr{K}_j(A, R_0)$, where X_0 is the initial guess, R_0 is the corresponding initial residual and

$$\mathscr{K}_j(A, R_0) = \text{BlockSpan}\left\{R_0, AR_0, \ldots, A^{j-1}R_0\right\} \subset \mathbb{C}^{n\times s}$$

is the jth block Krylov subspace related to A and R_0.

Generalized minimal residual (GMRES) (Saad & Schultz, 1986), conjugate gradient (CG) (Hermitian case) (Hestenes & Stiefel, 1952), CG squared (Sonneveld, 1989) and bi-CG stabilized (BiCGStab) (van der Vorst, 1992) are widely used Krylov subspace methods. They were all initially introduced in the simple case $s = 1$. An iteration of a simple Krylov method (i.e., $s = 1$) consists of a matrix–vector multiplication (BLAS2), dot products and update of vectors (BLAS1). In terms of high-performance computing these operations, especially the dot products, are constrained by communication (between processors or between levels of the local memory hierarchy) since the computation part becomes negligible when the number of processors increases. Thus, block-type Krylov methods were introduced. These schemes have three main advantages. First, matrix–set-of-vectors operations are used (BLAS3). Secondly, the solutions of multiple right-hand sides are computed simultaneously. Lastly, faster convergence can be achieved by using a larger search subspace. Generally, simple Krylov subspace methods have a block variant, (e.g., block GMRES (Vital, 1990), block BiCGStab (El Guennouni *et al.*, 2003) and block CG (O'Leary, 1980)). However, one issue related to block methods is that there are few papers addressing the convergence analysis, while for the methods previously mentioned, for the case $s = 1$, the literature is rich with such studies (Saad, 2003). O'Leary (1980) studies the convergence analysis of block CG and presents an estimation of the error in the approximate solutions. Simoncini & Gallopoulos (1996) generalize the theory of convergence presented in O'Leary (1980) to the block GMRES method. This generalization is restricted to the special case when the real part of the spectrum is positive definite.

The methods referred to as s-step methods, e.g., Chronopoulos & Gear (1989) and Hoemmen (2010) are based on the idea of performing s iterations of the simple method without communication (s here is different from the number of right-hand sides that we noted above). For this, s basis vectors are computed by performing s matrix–vector multiplications; then these basis vectors are orthogonalized by using block operations. The matrix–vector products are performed without communication. This is possible due to data redundancy. Recently, the enlarged Krylov subspace approach was introduced in Grigori *et al.* (2014a) along with a communication-reducing CG based on it. In order to enlarge the search subspace the authors split the initial residual into multiple vectors. They construct the block Krylov subspace that is associated to the matrix A and the block of vectors that is obtained from the splitting of the initial residual. The Krylov subspace is contained in the enlarged one. Thus, in the worst cases, it converges, at least, with the necessary iterations for CG to converge. In Grigori *et al.* (2014a) the authors present promising results when the enlarged Krylov CG is applied on linear systems that converge slowly with the simple CG method.

Iterative methods that rely on a block version of Krylov subspace produce inexact breakdowns, which are related to a rank deficiency in the block residual or in the block of search directions, before reaching convergence (Freund & Malhotra, 1997; Langou, 2003). Different strategies to deal with this issue are presented in the literature (Nikishin & Yeremin, 1995; Freund & Malhotra, 1997; Langou, 2003; Robbé & Sadkane, 2006; Gutknecht, 2007; Calandra *et al.*, 2013; Agullo *et al.*, 2014). To detect inexact breakdowns in block-like GMRES a rank test has to be done at each iteration. In Robbé & Sadkane (2006), Calandra *et al.* (2013) and Agullo *et al.* (2014) the authors propose an inexact breakdowns detection test based on Singular Value Decomposition (SVD) factorization of the block residual in the block Krylov subspace. This strategy implies a solution of a least squares problem at each iteration in order to obtain the block residual in the block Krylov subspace; then it performs its SVD factorization.

Downloaded from https://academic.oup.com/imajna/advance-article-abstract/doi/10.1093/imanum/dry054/5078636
by BIUS Jussieu user
on 28 August 2018

The dimension of the block Krylov residual increases linearly with the number of iterations. Different strategies to update a rank-revealing factorization exist in the literature. In Stewart (1993) and Li & Zeng (2005), given a rank-revealing factorization of a matrix M, the authors present how to update this factorization when several lines or columns are added, by concatenation, to the matrix M. In block Krylov methods the matrix $\bar{R}_j$ (the matrix that is factored to detect inexact breakdowns at iteration j) is different for each j. This matrix represents the block residual by the basis vectors of the block Krylov subspace, $R_j = V_{j+1}\bar{R}_j$. This representation changes from one iteration to another. More precisely, the matrix $\bar{R}_{j-1}$ is not necessarily a submatrix of $\bar{R}_j$. We reformulate the relation between R_j and the basis vectors of the block Krylov subspace by using an update strategy of the Hessenberg matrix; see Appendix A. This reformulation allows us to update the rank-revealing factorization and thus reduces the cost of the detection test.

Solving large-scale linear systems of equations by a long-recurrence Krylov method may require restarting the method. This slows down its convergence. To avoid this issue it is common to use deflation of eigenvalues (Morgan, 1995, 2005; Erhel *et al.*, 1996). Before restarting (block) GMRES either Ritz values or harmonic Ritz values and the associated vectors are computed to construct a deflation subspace.

In this paper we focus on the GMRES scheme as presented in Saad (2003). We introduce an enlarged GMRES method, referred to as EGMRES. This method is based on enlarged Krylov subspaces (Grigori *et al.*, 2014a). It is adapted for solving linear systems of equations with one or multiple right-hand sides. An *enlarging factor* EF is given as a parameter. At each iteration the dimension of the enlarged Krylov subspace increases by a number s_j between 1 and $s \times \mathrm{EF}$, where s is the number of right-hand sides. This number s_j decreases over iterations. The dimension of the enlarged Krylov subspace stops increasing when the exact solution is contained in the enlarged Krylov subspace. The enlarged Krylov subspace contains the classical (block) Krylov subspace. The EGMRES algorithm performs two global communication steps per iteration. The first corresponds to the orthogonalization of the new basis vectors against the previous vectors. The second is associated to the orthonormalization of the added basis vectors to the enlarged Krylov subspace. A point-to-point communication is necessary to perform the sparse matrix–matrix multiplication. Therefore, EGMRES and GMRES have the same number of messages per iteration. The size of the messages becomes larger in EGMRES. However, EGMRES converges faster especially on challenging linear systems; see Section 6. Thus, the number of global communication phases is reduced. In terms of arithmetic, EGMRES performs more floating point operations (flops) compared to GMRES. Nevertheless, we benefit from the efficiency of BLAS3 in performing these extra operations. In addition, this extra computation overlaps with communication when it is possible.

The paper is organized as follows. In Section 2 we give a brief discussion of existing variants of GMRES and its block version. We review exact and inexact breakdowns as introduced in Robbé & Sadkane (2006). We review the deflated Arnoldi procedure and the inexact breakdowns detection test that is proposed in Robbé & Sadkane (2006).

In Section 3 we introduce EGMRES. In Section 4 we present a strategy to reduce the size of the block basis vectors. We reformulate the detection test for inexact breakdowns presented first in Robbé & Sadkane (2006) and Calandra *et al.* (2013). This reformulation leads to the factorization of an $s \times s$ matrix rather than a matrix of dimension approaching $js \times s$, where s is the number of columns of the initial block residual R_0 and j is the iteration number. In addition, we show that this $s \times s$ matrix can be computed iteratively. Furthermore, we study a new strategy based on rank-revealing QR to reduce the size of the block in block GMRES (BGMRES) like methods. We show that the reduced basis is sufficient to achieve the same rate of convergence as when no reduction is done. We compare our strategy on a set of matrices to the existing approach that is based on SVD (Robbé & Sadkane, 2006), and we show that they have approximately the same behavior.

Downloaded from https://academic.oup.com/imajna/advance-article-abstract/doi/10.1093/imanum/dry054/5078636
by BIUS Jussieu user
on 28 August 2018

We show experimentally that the enlarged Krylov subspace method approximates the eigenvalues of the input matrix better than the classical GMRES method for the same basis size. This basis is built with a smaller number of iterations for the enlarged Krylov subspace method; hence, the cost of communication is less. We use this property to deflate eigenvalues between restart cycles. For this purpose we introduce a criterion based on both the approximated eigenvalue and the norm of the residual of the associated eigenvector.

We refer to the resulting method as restarted deflated EGMRES or RD-EGMRES. By using RD-EGMRES, we obtain a gain of a factor up to 7 with respect to GMRES in terms of iteration count on our set of matrices. We show numerically how the enlarged Krylov subspace method approximates the eigenvalues better.

In Section 5 we adapt EGMRES to be a constrained pressure residual (CPR) solver. The CPR-EGMRES is a special linear solver for saturation-pressure systems that arise from simulations of reservoirs. Since we are interested in solving linear systems arising from simulations of reservoirs we adapt EGMRES to be used as a CPR solver; the CPR solver was introduced in Wallis *et al.* (1985). Such linear systems are formed by two coupled systems. We propose to solve the global system (referred to as the second level) by using EGMRES. The first level corresponds to solving a subsystem associated with the pressure variable. This subsystem is solved at each iteration. To solve it, unlike the common choice of algebraic multigrid proposed in Cao *et al.* (2005), we introduce two practical strategies based on using RD-EGMRES. Thus, by using RD-EGMRES, we benefit from the approximation of eigenvectors to solve the linear system with multiple right-hand sides that are given one at a time. The first strategy uses a fixed number of iterations without the necessity of reaching the convergence threshold. The second strategy uses the threshold of convergence as a stopping criterion. Since a Krylov iterative method is not a linear operator in general, the first strategy requires the usage of a flexible variant in the global level. Note that the second strategy can be considered as a linear operator by reason of convergence (we suppose that the convergence threshold is small enough); hence, we do not need to use the flexible variant on the second level. We compare these strategies in the numerical experiments in Section 6. CPR-EGMRES reduces the number of iterations by up to a factor of 2 compared to the ideal CPR-GMRES that solves the first level with a direct LU solver.

Numerical experiments are presented in Section 6. First, we present results to show that the more we increase the enlarging factor, the faster the method converges. Furthermore, we show that reducing the basis by using the new strategy is as efficient as the approach based on SVD. We compare two thresholds for the criteria of eigenvalues deflation. This comparison is done with different maximal dimensions of the enlarged Krylov subspace. Then we show results for linear systems of equations with multiple right-hand sides, which are given one at a time. This is related to the CPR preconditioner that will be used later. Finally, results for linear systems of equations with multiple right-hand sides, given all at one time, are presented.

2. Background

In this section we review the block GMRES method, exact breakdowns and the deflated Arnoldi procedure.

2.1 *Notation*

MATLAB notation is used in a block sense: For a matrix M partitioned into block submatrices, $M(i,j)$ is the submatrix of M at block row i and block column j. We also refer to the block matrix M formed by the blocks $M(i,j)$ as $(M(i,j))_{i,j}$; $\|.\|_F$ represents the Frobenius norm. Let $t > 0$

Downloaded from https://academic.oup.com/imajna/advance-article-abstract/doi/10.1093/imanum/dry054/5078636
by BIUS Jussieu user
on 28 August 2018

be the enlarging factor of the (block) Krylov subspace and $T = ts$ the number of columns of the enlarged residual, where s is the number of right-hand sides. If $t = 1$ then the enlarged Krylov subspace is identical to the (block) Krylov subspace. Let $s_j \leqslant s$ be the number of added vectors to the basis of the block Krylov subspace $\mathscr{K}_{j-1}(A, R_0)$ at iteration j, and $c_j = s - s_j$. The dimension of the block Krylov subspace $\mathscr{K}_j(A, R_0)$ is $S_j = \sum_{i=1}^{j} s_i$. We denote the cardinal by #. The identity matrix of size l is denoted by I_l. The matrix of size $l \times m$ with zero elements is denoted by $0_{l,m}$. A tilde over a matrix V, i.e., $\tilde{V}$, means that an inexact breakdowns detection is done and this matrix is not updated yet. A bar over a matrix V, i.e., $\bar{V}$, is the representation of V in the projection subspace and this representation is by the constructed basis. The conjugate of V is represented by V^H and V^{T} represents the transpose of V. R_j and R_j^E are the (block) residual and the enlarged residual at the iteration j, respectively. Similarly, we note that X_j and X_j^E are the solution and the enlarged solution respectively. $\oplus$ refers to the direct sum between orthogonal subspaces. Finally, we define the following notation: $\tilde{V}_{j+1} \in \mathbb{C}^{n \times s_j}$ denotes the matrix whose columns are the generated basis vectors at iteration j; $V_{j+1} \in \mathbb{C}^{n \times s_{j+1}}$ is the matrix whose columns are effectively considered, as added vectors to the basis of $\mathscr{K}_j(A, R_0)$, to get $\mathscr{K}_{j+1}(A, R_0)$; $\mathscr{V}_j = [V_1, \ldots, V_j] \in \mathbb{C}^{n \times S_j}$ denotes the matrix whose columns are the basis vectors of the block Krylov subspace $\mathscr{K}_j(A, R_0)$; $D_j \in \mathbb{C}^{n \times c_{j+1}}$ is the matrix whose columns span the subspace left aside in iteration j.

2.2 *Block Arnoldi procedure and block GMRES*

The block Arnoldi procedure (see Algorithm 1) is the main part of the BGMRES method. It is basically the orthogonalization process applied on the new basis vectors to get an orthonormal basis for the block Krylov subspace.

Algorithm 1 Block-Arnoldi (A, V_1, m)

Require: orthogonal matrix $V_1 \in \mathbb{C}^{n \times s}$, matrix $A \in \mathbb{C}^{n \times n}$, number of iterations m.
Ensure: orthonormal block basis vector; $\mathscr{V}_{m+1}$, block Hessenberg matrix $H_m \in \mathbb{C}^{(m+1)s \times ms}$.

1: **for** $j = 1 : m$ **do**
2: $W = AV_j$.
3: **for** $i = 1 : j$ **do**
4: $H(i,j) = V_i^H W$.
5: **end for**
6: $W = W - \sum_{i=1}^{j} V_i H(i,j)$.
7: QR factorization of W, $W = V_{j+1} H(j+1, j)$.
8: $\mathscr{V}_m = [V_1, \ldots, V_m]$, $\mathscr{V}_{m+1} = [\mathscr{V}_m, V_{m+1}]$, $H_m = (H(i,j))_{i,j}$.
9: **end for**

The block GMRES method (see Algorithm 2), BGMRES (Vital, 1990), is a Krylov subspace method. It finds a sequence of approximate solutions $X_j, j > 0$ for the system of linear equations

Downloaded from https://academic.oup.com/imajna/advance-article-abstract/doi/10.1093/imanum/dry054/5078636
by BIUS Jussieu user
on 28 August 2018

$AX = B$. The residual norm $\|R_j\|_F$ is minimal over the corresponding block Krylov subspace

$$\mathscr{K}_j(A, R_0) = \text{BlockSpan}\{R_0, AR_0, \ldots, A^{j-1}R_0\}. \tag{2.1}$$

This method relies on building an orthonormal basis for the block Krylov subspace by using the block Arnoldi procedure. Once we build the basis we solve a linear least squares problem in that subspace to obtain the solution.

Algorithm 2 BGMRES

Require: matrix $A \in \mathbb{C}^{n\times n}$, right-hand sides $B \in \mathbb{C}^{n\times s}$, initial solution X_0 and the number of iterations m.
Ensure: approximate solution X_m.

1: $R_0 = B - AX_0 \in \mathbb{C}^{n\times s}$.
2: QR factorization of R_0, $R_0 = V_1\Pi_0$.
3: Get $\mathscr{V}_{m+1}$ and H_m using Block-Arnoldi (A, V_1, m) (Algorithm 1).
4: Solve the least squares problem $Y_m = \underset{Y\in\mathbb{C}^{ms\times s}}{\arg\min}\|H_mY - E_1\Pi_0\|_2$,
where $E_1 = (I_s, 0_{m,s})^{\mathrm{T}} \in \mathbb{C}^{js\times s}$.
5: $X_m = X_0 + \mathscr{V}_mY_m$.

An algebraic relation holds at each iteration of the algorithm, $AV_j = \sum_{i=1}^{j+1} V_iH_j(i,j)$. It leads to the relation

$$A\mathscr{V}_j = \mathscr{V}_jH_j(1:j, 1:j) + V_{j+1}H_j(j+1, j)E_j^{\mathrm{T}}, \tag{2.2}$$

where $\mathscr{V}_j = [V_1, \ldots, V_j]$, and $E_j = (0_{s,(j-1)s}, I_s)^{\mathrm{T}} \in \mathbb{C}^{js\times s}$. A detailed overview of block Krylov methods is given in Gutknecht (2007).

2.3 *Block Arnoldi and exact breakdown*

DEFINITION 2.1 A subspace $\mathscr{S} \subset \mathbb{C}^n$ is called **A-invariant** if it is invariant under multiplication by A, i.e., for all $u \in \mathscr{S}$, $Au \in \mathscr{S}$.

The importance of having an A-invariant subspace, for instance of dimension p, is that this subspace contains p exact eigenpairs if the matrix A is diagonalizable. In some cases, the matrix W, see (line 6, Algorithm 1), is rank deficient. This occurs when an A-invariant subspace is contained in the Krylov subspace.

DEFINITION 2.2 An exact breakdown (Robbé & Sadkane, 2006) is a phenomenon that occurs at the jth iteration in the block Arnoldi procedure when the matrix W, at (line 6, Algorithm 1), is rank deficient. The order of the exact breakdown at iteration j is the integer c_{j+1} verifying $c_{j+1} = s - \text{rank}(W)$ where s is the rank of V_1.

The following lemma is the GMRES case of Saad (2003, Proposition 6.1, p. 158). It illustrates the importance of the breakdown in GMRES; we give its proof for completeness.

Downloaded from https://academic.oup.com/imajna/advance-article-abstract/doi/10.1093/imanum/dry054/5078636
by BIUS Jussieu user
on 28 August 2018

LEMMA 2.3 In GMRES, when a breakdown occurs during iteration j, the Krylov subspace

$$\mathscr{K}_j(A, R_0) = \text{Span}\{V_1, \cdots, V_j\}$$

is an A-invariant subspace.

Proof. A breakdown in GMRES occurs during iteration j when $H_j(j+1, j) = 0$. Thus, immediately from relation (2.2), we get $A\mathscr{V}_j = \mathscr{V}_j H_j(1{:}j, 1{:}j)$. It yields that the subspace $\text{Span}\{V_1, \ldots, V_j\}$ is A-invariant.□

However, in general, for the block Arnoldi procedure an exact breakdown does not mean that there is an A-invariant subspace. For example, starting the algorithm with the initial block (u, Au), for any $u \in \mathbb{C}^n$, yields an exact breakdown in the first iteration. Nevertheless, the obtained subspace is not necessarily A-invariant. We recall several equivalent conditions related to the exact breakdown in Theorem 2.4. For the details and the proof see Robbé & Sadkane (2006). Let c_{j+1} denote the rank deficiency of W (line 6, Algorithm 1), i.e., $c_{j+1} = s - \text{rank}(W)$, where s is the rank of V_1.

THEOREM 2.4 In the block GMRES algorithm let X be the exact solution and R_j be the residual at iteration j. The conditions below are equivalent:

1. An exact breakdown of order c_{j+1} at iteration j occurs.
2. $\dim\{\text{Range}(V_1) \cap A\mathscr{K}_j(A, R_0)\} = c_{j+1}$.
3. $\text{rank}(R_j) = s - c_{j+1}$.
4. $\dim\{\text{Range}(X) \cap \mathscr{K}_j(A, R_0)\} = c_{j+1}$.

Proof. See the proof in Robbé & Sadkane (2006). □

As our method is based on the *enlarged Krylov subspace* it naturally inherits a block version of GMRES. In the next section we review the theory of the block GMRES method with deflation at each iteration proposed by Robbé & Sadkane (2006). This method was then reformulated in a different way by Calandra *et al.* (2013).

2.3.1 *Deflated Arnoldi relation.* Here we review the derivation of the modified algebraic relations of the Arnoldi procedure presented in, e.g., Robbé & Sadkane (2006) and Calandra *et al.* (2013). We follow the presentation in Calandra *et al.* (2013). We recall that $V_{j+1} \in \mathbb{C}^{n \times s_{j+1}}$ is the matrix formed by the columns considered to be useful and thus added to the basis $\mathscr{V}_j$ of the block Krylov subspace; $D_j \in \mathbb{C}^{n \times c_{j+1}}$ is the matrix whose columns span the useless subspace. The range of D_j is referred to as the deflated subspace. The decomposition of the range of the matrix $[\tilde{V}_{j+1}, D_{j-1}]$ into two subspaces is

$$\text{Range}([\tilde{V}_{j+1}, D_{j-1}]) = \text{Range}(V_{j+1}) \oplus \text{Range}(D_j), \tag{2.3}$$

with $[V_{j+1}, D_j]^H [V_{j+1}, D_j] = I_s$. The s_{j+1}-dimensional subspace, spanned by the columns of V_{j+1}, is added to the block Krylov subspace. The other c_{j+1}-dimensional subspace, spanned by D_j, is left aside. At the end of iteration j we want the following relation to hold:

$$A\mathscr{V}_j = [\mathscr{V}_{j+1}, D_j] H_j, \tag{2.4}$$

where the columns of D_j represent a basis of the deflated subspace after j iterations. The columns of $\mathscr{V}_{j+1}$ stand for a basis for the block Krylov subspace $\mathscr{K}_{j+1}$. We assume that this relation holds at the end

Downloaded from https://academic.oup.com/imajna/advance-article-abstract/doi/10.1093/imanum/dry054/5078636
by BIUS Jussieu user
on 28 August 2018

of iteration $j-1$. Thus,

$$A\mathscr{V}_{j-1} = [\mathscr{V}_j, D_{j-1}]H_{j-1}. \tag{2.5}$$

Let us study the iteration j. First, we multiply A by V_j. Then we orthogonalize against $\mathscr{V}_j$ and against D_{j-1}. A QR factorization of the result leads us to $\tilde{V}_{j+1}$. In matrix form that could be written in the following equation:

$$A\mathscr{V}_j = [\mathscr{V}_j, D_{j-1}, \tilde{V}_{j+1}]\tilde{H}_j, \tag{2.6}$$

where $\tilde{H}_j$ has the form

$$\tilde{H}_j = \begin{pmatrix} H_{j-1} & N_j \\ 0_{s_j, S_{j-1}} & M_j \end{pmatrix}, \tag{2.7}$$

$N_j = [\mathscr{V}_j, D_{j-1}]^H AV_j \in \mathbb{C}^{(S_{j-1}+s)\times s_j}$ and $(AV_j - [\mathscr{V}_j, D_{j-1}]N_j) = \tilde{V}_{j+1}M_j$ is the QR factorization. To transform the relation (2.6) to the form in (2.4) let $Q_{j+1} \in \mathbb{C}^{s\times s}$ be a unitary matrix such that

$$[D_{j-1}, \tilde{V}_{j+1}]Q_{j+1} = [V_{j+1}, D_j]; \tag{2.8}$$

then, we have

$$A\mathscr{V}_j = [\mathscr{V}_{j+1}, \mathscr{D}_j]\mathscr{Q}_{j+1}^H \tilde{H}_j, \tag{2.9}$$

where $\mathscr{Q}_{(j+1),j} = \begin{pmatrix} I_{S_j} & 0 \\ 0 & Q_{j+1} \end{pmatrix}$. Finally, we can write

$$A\mathscr{V}_j = [\mathscr{V}_{j+1}, D_j]H_j. \tag{2.10}$$

In conclusion the deflation of the converged subspace requires finding the matrix Q_{j+1}. We will address this later in Section 4.1. The strategy to reduce the basis is based on this algebra. In the remainder of this section we show the inexact breakdowns detection as presented in Robbé & Sadkane (2006) and Calandra *et al.* (2013).

2.4 *Inexact breakdowns and subspace decomposition*

In Robbé & Sadkane (2006) the authors introduce exact and inexact breakdowns in BGMRES-like methods. They define the inexact breakdown as the following.

DEFINITION 2.5 An inexact breakdown is a phenomenon that occurs when the matrix $(R_0\ AR_0\ \ldots\ A^mR_0)$ becomes almost rank deficient.

Detecting inexact breakdowns and deflating useless vectors lead to less computation and more memory for useful vectors. In Robbé & Sadkane (2006) they propose two strategies to detect inexact breakdowns. The first is related to the rank of the block residual while the second is related to the rank of the block basis vectors. In the same paper the analysis shows that in practice it is more likely to detect the rank deficiency of the block residual rather than the block basis vectors. In this paper we are interested in the detection test related to the block residual. Here, we present the inexact breakdowns detection test. We follow the presentation introduced in Calandra *et al.* (2013). We start from relation (2.8). Given the matrix $[D_{j-1}, \tilde{V}_{j+1}]$ and the block Krylov residual $\bar{R}_j \in \mathbb{C}^{(S_j+s)\times s}$ find Q_{j+1} such that V_{j+1} spans the

Downloaded from https://academic.oup.com/imajna/advance-article-abstract/doi/10.1093/imanum/dry054/5078636
by BIUS Jussieu user
on 28 August 2018

subspace that has not converged on the block residual. Let $\bar{R}_j = U\Sigma W^H$ be the SVD factorization of the block Krylov residual. In Robbé & Sadkane (2006) the authors decompose this factorization as

$$\begin{aligned}\bar{R}_j &= \begin{pmatrix} U_1 & U_2 \\ U_{s+} & U_{s-} \end{pmatrix} \begin{pmatrix} \Sigma_1 & \\ & \Sigma_2 \end{pmatrix} [W_1, W_2]^H \\ &= \begin{pmatrix} U_1 \\ U_{s+} \end{pmatrix} \Sigma_1 W_1^H + \begin{pmatrix} U_2 \\ U_{s-} \end{pmatrix} \Sigma_2 W_2^H, \end{aligned} \tag{2.11}$$

with $\|\Sigma_2\|_2 < \varepsilon_0$. The projection of the block residual $R_j \in \mathbb{C}^{n\times s}$ on the subspace perpendicular to $\mathscr{K}_j$ is given by

$$\begin{aligned}\big(I - \mathscr{V}_j\mathscr{V}_j^H\big)R_j &= [0, D_{j-1}, \tilde{V}_{j+1}]\bar{R}_j \\ &= [D_{j-1}, \tilde{V}_{j+1}]\big[U_{s+}\Sigma_1 W_1^H + U_{s-}\Sigma_2 W_2^H\big].\end{aligned}$$

The choice of the considered basis vectors from the linear combinations of the matrix $[D_{j-1}, \tilde{V}_{j+1}]$ relies on the idea that they should be related to the left singular vectors with singular values of Σ_1 or we can write

$$\text{Range}(V_{j+1}) = \text{Range}\big(\big(I - \mathscr{V}_j\mathscr{V}_j^H\big)R_jW_1\big) = \text{Range}\big([D_{j-1}, \tilde{V}_{j+1}]U_{s+}\Sigma_1\big).$$

To find the matrix Q_{j+1} it is sufficient to take the unitary factor of the QR factorization of $U_{s+} \in \mathbb{C}^{s\times s_j}$ and complete its columns to an orthonormal basis of $\mathbb{C}^{s\times s}$:

$$\begin{aligned}Q_{j+1} &= qr(U_{s+}\Sigma_1) \\ &= qr(U_{s+}).\end{aligned}$$

The detection test of inexact breakdowns is done at every iteration. Hence, an SVD factorization of $\bar{R}_j \in \mathbb{C}^{(s_j+s)\times s}$ occurs at each iteration. During a cycle the size of this problem grows linearly with the iteration number. In Stewart (1993) and Li & Zeng (2005) the authors update a rank-revealing factorization of a matrix M after adding several lines (or columns) to the matrix M by concatenation. However, the matrix $\bar{R}_j$ changes entirely from one iteration to another. In Section 4.1 we reformulate the relation between the jth block residual R_j and the basis vectors of the block Krylov subspace by using the update of the QR factorization of the Hessenberg matrix. As a consequence we obtain a rank-revealing update strategy that avoids solving the least squares problem in order to compute $\bar{R}_j$. Furthermore, a factorization of an $s \times s$ matrix is sufficient to detect the inexact breakdowns rather than performing an SVD factorization of a matrix of dimension approaching $js \times s$; see Proposition 4.2. In addition, a study of inexact breakdowns detection based on rank-revealing QR is presented.

3. EGMRES

We introduce in this section our new block GMRES method EGMRES, which is based on enlarging the block Krylov subspace (Grigori *et al.*, 2014a). Indeed, for each one of the s right-hand sides we add at each iteration multiple new basis vectors to the subspace. At the end the obtained search subspace contains the original block Krylov subspace. We describe briefly how the enlarged Krylov subspace is obtained by using projection operators, thus reformulating the derivation from Grigori *et al.* (2014a).

Downloaded from https://academic.oup.com/imajna/advance-article-abstract/doi/10.1093/imanum/dry054/5078636
by BIUS Jussieu user
on 28 August 2018

This method depends on the partition of the set of unknowns, which is obtained by partitioning the graph of the matrix by using K-way partitioning (Karypis & Kumar, 1998).

Let $\zeta = \{1, \ldots, n\}$. We partition this set in t disjoint nontrivial subsets denoted by (ζ_i) with $i = 1, \ldots, t$. To each subset we associate a projector P_i, such that

$$P_i : \mathbb{C}^{n\times s} \to \mathbb{C}^{n\times s}, \tag{3.1}$$

$$u \to Z_i Z_i^H u, \tag{3.2}$$

where the jth column of $Z_i \in \mathbb{R}^{n\times \#(\zeta_i)}$ is the $\zeta_i(j)$th canonical basis vector. Two properties follow:

$$u = \sum_{i=1}^{t} P_i(u) \quad \forall\, u \in \mathbb{C}^{n\times s}, \tag{3.3}$$

$$P_i \perp P_j, \quad i \neq j. \tag{3.4}$$

Before defining the enlarged Krylov subspace we define the enlarged residual using the projector P. We suppose that for all $i \in \{1, \ldots, t\}$ for all $j \in \{1, \ldots, s\}$, $\|P_i(R_0)(:, j)\|_2 \neq 0$. Thus, the enlarged residual is the matrix

$$P(R_0) = \big[P_1(R_0), \ldots, P_t(R_0)\big]. \tag{3.5}$$

Figure 1 illustrates the operator P; see relation (3.5).

DEFINITION 3.1 Let the system of linear equations $AX = B$, where $A \in \mathbb{C}^{n\times n}$ is nonsingular, $B \in \mathbb{C}^{n\times s}$ is full rank and $X \in \mathbb{C}^{n\times s}$. The jth t-enlarged Krylov subspace associated with the matrix A and the initial residual R_0 is defined by

$$\mathscr{K}_{j,t}(A, R_0) = \text{BlockSpan}\big\{P(R_0), AP(R_0), \ldots, A^{j-1}P(R_0)\big\}, \tag{3.6}$$

where the projection P is given by (3.5).

We will refer to the enlarged Krylov subspace $\mathscr{K}_{j,t}(A, R_0)$ by $\mathscr{K}_{j,t}$ when there is no ambiguity. For more details on the enlarged Krylov subspaces we refer the reader to Grigori *et al.* (2014a). While

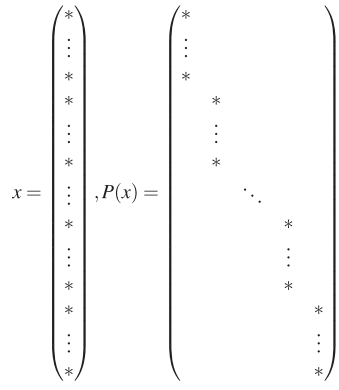

FIG. 1. Illustration of enlarging a vector x.

Downloaded from https://academic.oup.com/imajna/advance-article-abstract/doi/10.1093/imanum/dry054/5078636
by BIUS Jussieu user
on 28 August 2018

in Grigori *et al.* (2014a) the authors focused on Symmetric Positive Definite (SPD) matrices, in the following we derive an enlarged Krylov subspace method for solving linear systems of equations of general matrices.

DEFINITION 3.2 (EGMRES). EGMRES is an enlarged Krylov subspace method. It finds a sequence of approximate solutions $\{X_1, \ldots, X_m\}$ for the system of linear equations $AX = B$, where $X_j - X_0$ belongs to the jth enlarged Krylov subspace $\mathscr{K}_{j,t}(A, R_0)$ with R_0 the initial residual and $\|R_j\|_F = \|B - AX_j\|_F$ is minimal over the enlarged Krylov subspace.

3.1 *EGMRES algorithm*

The following algorithm is the basic form of EGMRES. Let $\mathbf{1}_t$ be as

$$\mathbf{1}_t = [I_s, \ldots, I_s]^{\mathrm{T}} \in \mathbb{C}^{T \times s}, \tag{3.7}$$

where I_s the identity matrix of size s.

The update of the Hessenberg matrix (line 8, Algorithm 3) means updating its QR factors $\mathscr{F}_j$ and C_j such that $H_j = \mathscr{F}_j \begin{pmatrix} C_j \\ 0_{T,jT} \end{pmatrix}$, where $\mathscr{F}_j \in \mathbb{C}^{(j+1)T \times (j+1)T}$ and $C_j \in \mathbb{C}^{jT \times jT}$.

In the following we prove that the EGMRES method finds the approximate solution X_j at iteration j such that the residual R_j has minimal Frobenius norm over the enlarged Krylov subspace $\mathscr{K}_{j,t}(A, R_0)$.

PROPOSITION 3.3 Following the notation in Algorithm 3 we have

$$\|B - AX_j\|_F = \min_{Y \in \mathbb{C}^{jT \times T}} \|\Pi_j \mathbf{1}_t - H_j Y \mathbf{1}_t\|_F.$$

Proof. We have

$$\begin{aligned} \left\|B - A(X_j + X_0)\right\|_F &= \left\|R_0^E \mathbf{1}_t - AX_j\right\|_F \\ &= \left\|V_1 \Pi_0 \mathbf{1}_t - [V_1, \ldots, V_{j+1}] H_j Y_j \mathbf{1}_t\right\|_F \\ &= \|\Pi_j \mathbf{1}_t - H_j Y_j \mathbf{1}_t\|_F. \end{aligned}$$

By construction Y_j minimizes the Frobenius norm of $\|\Pi_j \mathbf{1}_t - H_j Y \mathbf{1}_t\|_F$, where $Y \in \mathbb{C}^{jT \times T}$. Thus,

$$\left\|B - A(X_j + X_0)\right\|_F = \min_{Y \in \mathbb{C}^{jT \times T}} \|\Pi_j \mathbf{1}_t - H_j Y \mathbf{1}_t\|_F.$$

□

After the presentation of the EGMRES method we remark that once we enlarge the block residual it returns to the block GMRES scheme. More precisely, let $b \in \mathbb{R}^{n \times s}$ be a set of vectors. In exact arithmetic the operations at iteration i of EGMRES when solving $Ax = b$ are the same as the operations performed at iteration i of block GMRES when solving $Ax = P(b)$, where P is the enlarging operator that is defined in relation (3.5). Two differences exist between these two algorithms. The stopping criterion changes as shown in Proposition 3.3. EGMRES achieves convergence when the norm of the enlarged residual multiplied by the matrix $\mathbf{1}_t$, defined in relation (3.7), is less than the convergence threshold. In block GMRES the convergence is achieved when each vector of the block residual has a norm less than

Downloaded from https://academic.oup.com/imajna/advance-article-abstract/doi/10.1093/imanum/dry054/5078636
by BIUS Jussieu user
on 28 August 2018

Algorithm 3 EGMRES

Require: threshold of convergence ε_0, initial solution X_0.
Ensure: approximate solution X_j.

1: $R_0 = B - AX_0$.
2: Form the enlarged residual $P(R_0)$ as in (3.5).
3: $R_0^E = P(R_0)$.
4: QR factorize R_0^E, $R_0^E = V_1 \Pi_0$.
5: Set $E_0 = \Pi_0$ and $G_0 = 0_{T,T}$.
6: **for** $j = 1$ till convergence **do**
7: $W = AV_j$.
8: Orthogonalization procedure to get V_{j+1} and updating H_j and its QR factors $H_j = \mathscr{F}_j \begin{pmatrix} C_j \\ 0_{T,jT} \end{pmatrix}$ where $\mathscr{F}_j \in \mathbb{C}^{(j+1)T \times (j+1)T}$ and $C_j \in \mathbb{C}^{jT \times jT}$.
9: Compute $\begin{pmatrix} E_j \\ G_j \end{pmatrix} = \mathscr{F}_j^H \Pi_j$ where $\Pi_j = \begin{pmatrix} \Pi_0 \\ 0_{jT,T} \end{pmatrix}$, $E_j \in \mathbb{C}^{jT \times T}$ and $G_j \in \mathbb{C}^{T \times T}$.
10: **if** $\|G_j \mathbf{1}_t\|_F < \varepsilon_0$ **then**
11: Break.
12: **end if**
13: **end for**
14: Solve the linear least squares problem $Y_j = \underset{Y \in \mathbb{C}^{jT \times T}}{\arg\min} \|\Pi_j - H_j Y\|$, $Y_j = C_j \setminus E_j$.
15: $X_j = X_0 + [V_1, \ldots, V_j] Y_j \mathbf{1}_t$.

the convergence threshold. We note that the stopping criterion of block GMRES is stronger than the stopping criterion of EGMRES. The solution in EGMRES is recovered by multiplying the vectors of x by $\mathbf{1}_t$, while the solution vectors in block GMRES correspond to the vectors of right-hand sides.

4. Inexact breakdowns and eigenvalues deflation in block GMRES

As mentioned previously EGMRES and block GMRES do the same operations during each iteration if we consider the enlarged residual as an initial block residual. For this, in this section we study block GMRES rather than EGMRES. The application of this study on EGMRES is direct.

Relations in Lemma 4.1 and Proposition 4.2 hold until the end of this paper.

Downloaded from https://academic.oup.com/imajna/advance-article-abstract/doi/10.1093/imanum/dry054/5078636
by BIUS Jussieu user
on 28 August 2018

4.1 *Inexact breakdowns*

In this section we reformulate the inexact breakdowns detection test in order to reduce its cost. In Robbé & Sadkane (2006) and Calandra *et al.* (2013) the authors propose a strategy to detect inexact breakdowns related to the block residual. We showed in Section 2 how this strategy adds only useful vectors to the block Krylov subspace. However, it needs to do an SVD factorization of the matrix representing the block Krylov residual $\bar{R}_j \in \mathbb{C}^{(S_j+s)\times s}$. Rank-revealing update strategies exist in the literature (Stewart, 1993; Li & Zeng, 2005). When several lines (or columns) are concatenated to a matrix M an update of its rank-revealing factorization is not costly. However, the block residual in block GMRES changes entirely every iteration and has a dimension that depends on the iteration number j. Proposition 4.2 is the key idea to reduce the dimension of the SVD problem. Before that we need the following Lemma 4.1. This lemma is going to be a tool in the remainder of this section.

LEMMA 4.1 The QR factorization of the matrix $\tilde{H}_j$ in the relation (2.6) is given by the relation

$$\tilde{H}_j = \left(\prod_{i=0}^{j-2} \mathcal{Q}_{(j-i),j}^H\right)\left(\prod_{i=1}^{j} \mathcal{F}_{i,j}\right)\begin{pmatrix} C_j \\ 0_{s,S_j}\end{pmatrix}, \tag{4.1}$$

where

$$\mathcal{Q}_{i,j} = \begin{pmatrix} I_{S_{i-1}} & & \\ & Q_i & \\ & & I_{(S_j-S_{i-1})}\end{pmatrix}, \ \mathcal{F}_{i,j} = \begin{pmatrix} I_{S_{i-1}} & & \\ & F_i & \\ & & I_{(S_j-S_i)}\end{pmatrix}$$

and $C_j \in \mathbb{C}^{S_j\times S_j}$ is triangular. The rotation matrix obtained by the inexact breakdowns test is $Q_i \in \mathbb{C}^{s\times s}$ and $F_i \in \mathbb{C}^{(s_i+s)\times(s_i+s)}$ is the Householder transformation matrix used to triangularize the block $\tilde{H}_i(i : i+1, i)$ after updating $\tilde{H}_i(1 : i, i)$ by using F_k for $k = 1, \ldots, i-1$. By convention $S_0 = 0$.

Proof. The proof is constructive and is given in the appendix. □

PROPOSITION 4.2 The following relations hold during block GMRES with the Arnoldi procedure.

1. $R_0 = [\mathcal{V}_j, D_{j-1}, \tilde{V}_{j+1}]\left(\prod_{i=0}^{j-1} \mathcal{Q}_{(j-i),j}^H\right)\begin{pmatrix}\Pi_0 \\ 0_{S_j,s}\end{pmatrix}$.
2. $\|B - A(X_0 + \mathcal{V}_j Y)\|_F = \left\|\left(\prod_{i=0}^{j-1} \mathcal{F}_{(j-i),j}^H\right)\mathcal{Q}_{1,j}^H\begin{pmatrix}\Pi_0 \\ 0_{S_j,s}\end{pmatrix} - \begin{pmatrix} C_j \\ 0_{s,S_j}\end{pmatrix} Y\right\|_F$.
3. $Y_j = C_j \backslash E_j$.
4. $R_j = [\mathcal{V}_j, D_{j-1}, \tilde{V}_{j+1}]\left(\prod_{i=0}^{j-2} \mathcal{Q}_{(j-i),j)}^H\right)\left(\prod_{i=1}^{j} \mathcal{F}_{i,j}\right)\begin{pmatrix}0_{S_j,s} \\ G_j\end{pmatrix}$.

Where Π_0 verifies the relation $R_0 = \tilde{V}_1\Pi_0$, $Y \in \mathbb{C}^{S_j\times s}$ and

$$\begin{pmatrix} E_j \\ G_j\end{pmatrix} = \left(\prod_{i=0}^{j-1} \mathcal{F}_{(j-i),j}^H\right)\mathcal{Q}_{1,j}^H\begin{pmatrix}\Pi_0 \\ 0_{S_j,s}\end{pmatrix}$$

such that $E_j \in \mathbb{C}^{S_j\times s}$ and $G_j \in \mathbb{C}^{s\times s}$.

Downloaded from https://academic.oup.com/imajna/advance-article-abstract/doi/10.1093/imanum/dry054/5078636
by BIUS Jussieu user
on 28 August 2018

Proof. Proof is by induction for 1, and it is immediate for the rest. □

In the block GMRES method a linear combination of the block residual could converge, while the system has not converged yet. This leads to unnecessary computations and memory loss. To remedy this issue we use a deflation technique based on detection of inexact breakdowns.

Robbé & Sadkane (2006) introduced two criteria based on SVD to determine the convergent subspace. The first depends on the block residual. The second depends on the block basis vector. In a later paper Calandra *et al.* (2013) reformulated the first criterion with a slight modification, leading to a different least squares problem.

The detection test is based on an SVD factorization of a matrix of size $(S_j + s) \times s$ at iteration j. This cost depends on the iteration number and it becomes expensive quickly. We propose in the next section a new strategy to reduce the problem to a matrix of size $s \times s$; hence, the cost becomes independent of iteration. Moreover, we also study the detection of a test based on rank-revealing QR.

4.2 *Inexact breakdowns detection*

Here we present the reformulation of the inexact breakdowns test that is presented in Robbé & Sadkane (2006) and Calandra *et al.* (2013). This reformulation leads to a reduction of the dimension of the SVD test that is used to detect inexact breakdowns in the block residual of block GMRES. This theory can be applied to all block GMRES-like methods.

PROPOSITION 4.3 An SVD factorization on the matrix G_j is equivalent to an SVD factorization of $\bar{R}_j$.

Proof. Proposition 4.2 proves that

$$\bar{R}_j = \left(\prod_{i=0}^{j-2} \mathcal{Q}_{(j-i),j}^H \right) \left(\prod_{i=1}^{j} \mathcal{F}_{i,j} \right) \begin{pmatrix} 0_{S_j,s} \\ G_j \end{pmatrix}. \tag{4.2}$$

Let $G_j = U \Sigma W^H$ be the SVD factorization of G_j. Since $\left(\prod_{i=0}^{j-2} \mathcal{Q}_{(j-i),j}^H \right) \left(\prod_{i=1}^{j} \mathcal{F}_{i,j} \right)$ is unitary we find that

$$\bar{R}_j = \left(\prod_{i=0}^{j-2} \mathcal{Q}_{(j-i),j}^H \right) \left(\prod_{i=1}^{j} \mathcal{F}_{i,j} \right) \begin{pmatrix} 0_{S_j,s} \\ U \end{pmatrix} \Sigma W^H$$

is an SVD factorization of $\bar{R}_j$ with $\left(\prod_{i=0}^{j-2} \mathcal{Q}_{(j-i),j}^H \right) \left(\prod_{i=1}^{j} \mathcal{F}_{i,j} \right) \begin{pmatrix} 0_{S_j,s} \\ U \end{pmatrix}$ standing for the left unitary factor. □

COROLLARY 4.4 A rank-revealing QR factorization on the matrix G_j is equivalent to a rank-revealing QR factorization of $\bar{R}_j$.

Proof. The proof is similar to the proof of Proposition 4.3. □

Downloaded from https://academic.oup.com/imajna/advance-article-abstract/doi/10.1093/imanum/dry054/5078636
by BIUS Jussieu user
on 28 August 2018

An important difference related to the reference test presented in Robbé & Sadkane (2006) and Calandra *et al.* (2013) is that the dimension of the factored matrix does not depend on the iteration number j. In the mentioned references this dimension is $S_j \times s$ at iteration j. Proposition 4.3 shows that this dimension is minimal.

In addition, it shows that there is no need to compute the residual $\bar{R}_j$ at each iteration to detect inexact breakdowns. Indeed, to get the matrix G_j, it is sufficient to update E_j, G_j, by using F_j. This matrix G_j is computed at each iteration in order to perform the stopping criterion. Thus, there is no need to solve the least squares problem entirely.

In the remainder of this section we introduce an inexact breakdowns detection based on rank-revealing QR to reduce the cost of performing an SVD factorization.

We start from relation (2.8). Given the matrix $[D_{j-1}, \tilde{V}_{j+1}]$ and the matrix $G_j \in \mathbb{C}^{s\times s}$ that verifies the relation

$$\begin{pmatrix} 0_{S_j,s} \\ G_j \end{pmatrix} = \left(\prod_{i=0}^{j-1} \mathscr{F}^H_{(j-i),j} \right) \left(\prod_{i=2}^{j} \mathscr{Q}_{i,j} \right) \bar{R}_j$$

as presented in Proposition 4.2, find Q_{j+1} such that V_{j+1} spans the subspace related to the nonconvergent part of the block residual.

In Robbé & Sadkane (2006) and Calandra *et al.* (2013) the authors propose a strategy based on the SVD of the matrix $\bar{R}_j \in \mathbb{C}^{(S_j+s)\times s}$. The detection test of the inexact breakdowns is done at every iteration. Hence, an SVD factorization of $\bar{R}_j \in \mathbb{C}^{(S_j+s)\times s}$ occurs at each iteration. During a cycle, the size of this problem grows linearly with the iteration number. We propose a new strategy to keep the dimension of the SVD problem constant and equal to $s \times s$. Furthermore, using rank-revealing QR factorization (Chan, 1987) instead of SVD factorization reduces the computational complexity. Here we derive the theory of that strategy.

Let ε_0 be a threshold given and $G_j = S\mathscr{R}P^{\mathrm{T}}$ be a rank-revealing QR factorization of the matrix $G_j \in \mathbb{C}^{s\times s}$. The matrix S stands for an orthonormal basis for the range of G_j, $\mathscr{R}$ is an upper triangular matrix and P is a permutation matrix. We can write the rank-revealing QR relation in the form

$$\begin{aligned} G_j &= \begin{pmatrix} S_+ & S_- \end{pmatrix} \begin{pmatrix} \mathscr{R}_1 & \mathscr{R}_2 \\ 0_{c_{j+1},s_{j+1}} & \mathscr{R}_3 \end{pmatrix} \begin{pmatrix} P_1^{\mathrm{T}} \\ P_2^{\mathrm{T}} \end{pmatrix} \\ &= S_+ \begin{pmatrix} \mathscr{R}_1 & \mathscr{R}_2 \end{pmatrix} \begin{pmatrix} P_1^{\mathrm{T}} \\ P_2^{\mathrm{T}} \end{pmatrix} + S_- \mathscr{R}_3 P_2^{\mathrm{T}}, \end{aligned} \tag{4.3}$$

with $\|\mathscr{R}_3\|_2 < \varepsilon_0$. Directly, we have that s_{j+1} is the numerical rank of G_j, i.e., the number of columns in S_+.

To detect inexact breakdowns by using rank-revealing QR (RRQR) the test depends on the same idea that is proposed in Robbé & Sadkane (2006) and Calandra *et al.* (2013). The new basis vectors to be added should be related to the subspace that has not converged on the range of $\bar{R}_j$. We write the projection of the residual on the subspace perpendicular to $\mathscr{K}_j$ using the RRQR decomposition.

Downloaded from https://academic.oup.com/imajna/advance-article-abstract/doi/10.1093/imanum/dry054/5078636
by BIUS Jussieu user
on 28 August 2018

Note that $\mathscr{S}\mathscr{S}^H R_j = R_j$ where

$$\mathscr{S} = [\mathscr{V}_j, D_{j-1}, \tilde{V}_{j+1}] \left(\prod_{i=0}^{j-2} \mathscr{Q}^H_{(j-i),j)} \right) \left(\prod_{i=1}^{j} \mathscr{F}_{i,j} \right) \begin{pmatrix} 0_{S_j,s} \\ S \end{pmatrix},$$

$$\left(I - \mathscr{V}_j \mathscr{V}_j^H\right) R_j = [0, D_{j-1}, \tilde{V}_{j+1}] \bar{R}_j \tag{4.4}$$

$$= [0, D_{j-1}, \tilde{V}_{j+1}] \left(\prod_{i=0}^{j-2} \mathscr{Q}^H_{(j-i),j)} \right) \left(\prod_{i=1}^{j} \mathscr{F}_{i,j} \right) \begin{pmatrix} 0_{S_j,s} \\ S \end{pmatrix} \mathscr{R} P^{\mathrm{T}}. \tag{4.5}$$

We want $\mathrm{Range}(V_{j+1}) = \mathrm{Range}((I - \mathscr{V}_j \mathscr{V}_j^H)\mathscr{S}_+ \mathscr{S}_+^H R_j)$, where $\mathscr{S}_+$ is the first s_{j+1} columns of $\mathscr{S}$. Thus,

$$\mathrm{Range}(V_{j+1}) = \mathrm{Range}\left([0, D_{j-1}, \tilde{V}_{j+1}] \left(\prod_{i=0}^{j-2} \mathscr{Q}^H_{(j-i),j)} \right) \left(\prod_{i=1}^{j} \mathscr{F}_{i,j} \right) \begin{pmatrix} 0_{S_j,s} \\ S_+ \end{pmatrix} \mathscr{R} P^{\mathrm{T}} \right)$$

$$= \mathrm{Range}\left([D_{j-1}, \tilde{V}_{j+1}] \underline{S} \right). \tag{4.6}$$

The unitary factor of the QR factorization of $\underline{S}$ is Q_{j+1}. As a result we have

$$\mathrm{Range}(V_{j+1}) \oplus \mathrm{Range}(D_j) = \mathrm{Range}(\tilde{V}_{j+1}) \oplus \mathrm{Range}(D_{j-1}).$$

The columns of the matrix D_j form a subspace of the block Krylov subspace. They are chosen in the best way so that the newly added basis vectors V_{j+1} are optimal. In fact, V_{j+1} helps to minimize only the largest singular values of the residual block in the next iteration. A threshold is given to separate the largest and the smallest singular values. Thus, the smallest singular values are neglected.

Algorithm 4 shows how to compute the matrix Q_{j+1}; see (2.9).

Algorithm 4 Inexact breakdowns detection(G_j, ε)

Require: $G_j \in \mathbb{C}^{s \times s}$ and ε the tolerance of inexact breakdown.
Ensure: Q_{j+1} and s_{j+1}.

1: RRQR factorization of G_j, $G_j = S\mathscr{R}P^{\mathrm{T}}$.

2: $G_j = S_+(\mathscr{R}_1 P_1^{\mathrm{T}} + \mathscr{R}_2 P_2^{\mathrm{T}}) + S_-\mathscr{R}_3 P_2^{\mathrm{T}}$, where $\mathscr{R}_3$ has maximum size such that its second norm is less than ε. The rank of G_j is s_{j+1}.

3: QR factorization of $\underline{S}$ (4.6), Q_{j+1} is the unitary factor.

4.3 *Deflation of eigenvalues*

As the size of the memory is limited we normally need to use the restart variant by disregarding all the built block Krylov subspaces and rebuilding a new one beginning with the last residual. This means a loss of information. A common approach to keeping useful information is to deflate small eigenvalues

Downloaded from https://academic.oup.com/imajna/advance-article-abstract/doi/10.1093/imanum/dry054/5078636
by BIUS Jussieu user
on 28 August 2018

if their eigenvectors have converged or if they are well approximated by the end of the cycle (Morgan, 1995; Erhel *et al.*, 1996; Tang *et al.*, 2009; Grigori *et al.*, 2014b).

In the remainder of this section we show how these eigenvalues and eigenvectors are chosen. We first recall a theorem from Erhel *et al.* (1996). The algebraic formulation of the eigenvalues deflation preconditioner follows its result. We reformulated the theorem to make it conform with the context of the paper.

THEOREM 4.5 Suppose that the matrix A is diagonalizable and let $\{\lambda_1, \ldots, \lambda_n\}$ be the set of eigenvalues of A with $|\lambda_1| \leqslant \cdots \leqslant |\lambda_n|$ and $\{u_1, \ldots, u_n\}$ be the corresponding normal eigenvectors. Given a threshold ε_1, set m to be the positive integer such that $|\lambda_m| < \varepsilon_1 \leqslant |\lambda_{m+1}|$. Let $U = \big(u_1, \ldots, u_m\big) = ZL$ be the QR factorization and $M = I_n + Z\big(\frac{1}{|\lambda_n|} Z^H A Z - I_m\big) Z^H$; then

1. the matrix M is invertible and its inverse $\tilde{M} = I_n + Z(|\lambda_n|(Z^H A Z)^{-1} - I_m) Z^H$;
2. the matrix AM^{-1} has eigenvalues $\{\lambda_n, \ldots, \lambda_n, \lambda_{m+1}, \ldots, \lambda_n\}$ with $\{u_1, \ldots, u_n\}$ as corresponding eigenvectors.

Proof. A similar proof using invariant subspaces is given in Erhel *et al.* (1996). □

4.3.1 *Well-approximated eigenvalues and eigenvectors.* Let S_1 be the upper $k_{\text{cycle}} \times k_{\text{cycle}}$ submatrix of the matrix H_{cycle}, where cycle is the number of the last iteration in block GMRES. Suppose that S_1 is diagonalizable (it is sufficient to suppose that so is A), and let $\{\lambda_1, \ldots, \lambda_m\}$ be the eigenvalues of S_1 with absolute value less than a given threshold, ε_1. In Algorithm 5 we propose an approach to measure the approximated eigenvector residual norm. No multiplication by the matrix A is necessary. Actually, we just need the matrix H_{cycle} and the eigenvector u in the block Krylov subspace to perform the test. The following theorem addresses the theoretical part that Algorithm 5 depends on.

PROPOSITION 4.6 Let H_{cycle} be the matrix verifying

$$A\mathscr{V}_{\text{cycle}} = [\mathscr{V}_{\text{cycle}+1}, \mathscr{D}_{\text{cycle}}] H_{\text{cycle}}$$

and denote by S_1 the maximal square submatrix of H_{cycle} obtained by deleting lines from the bottom of H_{cycle}, such that $H_{\text{cycle}} = \begin{pmatrix} S_1 \\ S_2 \end{pmatrix}$. Let u be an eigenvector of the matrix S_1 with eigenvalue λ. Then

$$\|A\mathscr{V}_{\text{cycle}} u - \lambda \mathscr{V}_{\text{cycle}} u\|_2 = \|S_2 u\|_2.$$

Proof.

$$\begin{aligned} A\mathscr{V}_{\text{cycle}} u &= [\mathscr{V}_{\text{cycle}+1}, D_{\text{cycle}}] H_{\text{cycle}} u \\ &= [\mathscr{V}_{\text{cycle}+1}, D_{\text{cycle}}] \begin{pmatrix} S_1 \\ S_2 \end{pmatrix} u \\ &= \mathscr{V}_{\text{cycle}} S_1 u + [V_{\text{cycle}+1}, D_{\text{cycle}}] S_2 u \\ &= \lambda \mathscr{V}_{\text{cycle}} u + [V_{\text{cycle}+1}, D_{\text{cycle}}] S_2 u. \end{aligned}$$

Downloaded from https://academic.oup.com/imajna/advance-article-abstract/doi/10.1093/imanum/dry054/5078636
by BIUS Jussieu user
on 28 August 2018

Since $[V_{\text{cycle}+1}, D_{\text{cycle}}]$ is unitary it yields

$$\|A\mathscr{V}_{\text{cycle}}u - \lambda\mathscr{V}_{\text{cycle}}u\|_2 = \|S_2u\|_2.$$

□

Scaling the spectrum of the linear system is taken into account in practice. To decide if a vector $\mathscr{V}_{\text{cycle}-1}u$ is a good approximation of an eigenvector we compute $\frac{\|S_2u\|_2}{|\lambda_{\max}|}$. If it is less than the given threshold this vector will be deflated. A good approximation of $|\lambda_{\max}|$ is computed after the first restart, since this eigenvalue converges fast.

Algorithm 5 Deflation of eigenvalues $(A, V_{\text{cycle}}, H, Z, |\lambda_{\max}|, \varepsilon, \text{nev})$

Require: the matrix A, the basis of the block Krylov subspace $\mathscr{V}_{\text{cycle}}$, the matrix $H = \begin{pmatrix} S_1 \\ S_2 \end{pmatrix}$, threshold for convergence of eigenvalues and eigenvectors ε, the maximum number of eigenvalues to deflate nev, already deflated eigenvectors Z (optional), $|\lambda_{\max}|$ (optional).

Ensure: nev, and the terms used in the preconditioner Z and Z^HAZ, an approximation of the magnitude of the largest eigenvalue $|\lambda_{\max}|$ (optional).

1: Calculate the eigenvalues $\{\lambda_1, \ldots, \lambda_p\}$ of S_1 with absolute value less than ε and their corresponding normal eigenvectors $\{u_1, \ldots, u_p\}$.
2: Set $U = [\,]$.
3: **if** $|\lambda_{\max}|$ is not provided **then**
4: Compute $|\lambda_{\max}|$.
5: **end if**
6: **for** $i = 1 : \min\{p, \text{nev}\}$ **do**
7: **if** $\|S_2u_i\|_2 < |\lambda_{\max}|\varepsilon$ **then**
8: $U = [U, u]$. /* *Deflate* λ_i */
9: nev = nev − 1.
10: **end if**
11: **end for**
12: QR factorization of U, $U = ZL$.
13: Expand the vectors of Z to $\mathbb{C}^n$, $Z = \mathscr{V}_{\text{cycle}}Z$.
14: Form $Z^HAZ = L^H\Lambda L$ to use in the preconditioner, Λ is a diagonal matrix with the deflated λ_i.

4.4 *RD-BGMRES(m)*

In Algorithm 6 we present the *restarted deflated BGMRES(m)*, where m is the maximum number of vectors to be saved in memory, including both basis vectors and approximated eigenvectors. This algorithm is the main part of the CPR-EGMRES method. It is applied at each iteration of the CPR solver as presented in the following section.

Downloaded from https://academic.oup.com/imajna/advance-article-abstract/doi/10.1093/imanum/dry054/5078636
by BIUS Jussieu user
on 28 August 2018

Algorithm 6 RD-BGMRES(m)

Require: the matrix A, the right-hand side B, the initial guess X_0, the tolerance of convergence ε_0, the tolerance of eigenvalues and eigenvector residual norm ε_1, the maximal number of deflated eigenvalues $\text{nev}_{\max}$, the maximal number of cycles $cycle_{\max}$, the maximal number of vectors to be saved in memory m, preconditioner M (I if not given).

Ensure: the approximate solution X_a of the system $AX = B$, the preconditioner M.

Set $\text{nev} = \text{nev}_{\max}$, $R_0 = B - AX_0$.
for $\text{cycle} = 1 : \text{cycle}_{\max}$ **do**
  if cycle > 1 and nev > 0 **then**
    Call Algorithm 5 to obtain Z.
    Update the preconditioner M^{-1} and update nev.
  end if
  QR factorization of R_0, $R_0 = \tilde{V}_1 \Pi_0$.
  Call Algorithm 4 (Π_0, ε) to determine the matrix Q_1 and s_1. $S_1 = s_1$.
  Set $E_0 = Q_1 \Pi_0$, $G_0 = 0_{s_1,s}$.
  $[V_1, D_0] = \tilde{V}_1 Q_1$, with $V_1 \in \mathbb{C}^{n \times s_1}$ and $D_0 \in \mathbb{C}^{n \times s - s_1}$. Set $j = 0$.
  while $S_{j+1} + s < m$ **do**
    Set $j = j + 1$. $W = AM^{-1}V_j$.
    Orthogonalize W against $\mathcal{V}_j$ and D_{j-1}.
    QR factorize W. Build $\tilde{V}_{j+1}$ and get $\mathcal{F}_{j,j}$ to update the QR factorization of $\tilde{H}_j$ as in (4.1).
    $$\begin{pmatrix} E_j \\ G_j \end{pmatrix} = \mathcal{F}_{j,j}^H \begin{pmatrix} E_{j-1} \\ G_{j-1} \\ 0_{s_j,s} \end{pmatrix}.$$
    Call Algorithm (4) (G_j, ε_1) to determine the matrix $\mathcal{Q}_{j+1,j}$ and s_{j+1}.
    $S_{j+1} = S_j + s_{j+1}$.
    if $\|G_j\|_F < \varepsilon_0$ **then**
      Break.
    end if
  end while
  $Y_j = C_j \setminus E_j$. $X_j = M^{-1}\mathcal{V}_j Y_j$. $X_a = X_0 + X_j$.
  $R_j = R_0 - AX_j$. Set $R_0 = R_j$, and $X_0 = X_a$.
end for

We note that the parameters $\varepsilon_0, \varepsilon_1$ in Algorithm 6 are not related; see Table 3 and the discussion in Section 6.

Downloaded from https://academic.oup.com/imajna/advance-article-abstract/doi/10.1093/imanum/dry054/5078636
by BIUS Jussieu user
on 28 August 2018

5. CPR-EGMRES

In this section, we introduce the CPR preconditioner with EGMRES. This preconditioner was first introduced by Wallis *et al.* (1985) as a preconditioner for the solution of systems of linear equations arising from the simulations of reservoirs. In simulations of reservoirs the overall system is of mixed character. However, the pressure field usually has a near elliptic behavior with long-range coupling, while the remaining equations (referred to as saturation equations) often possess near hyperbolic character with steep local gradients (Wallis *et al.*, 1985; Cao *et al.*, 2005). As a direct consequence the linear systems in simulations of reservoirs are a natural target for a two-stage preconditioning strategy.

5.1 *Two-stage preconditioning*

The two-stage preconditioning formula is given by

$$M_{1,2}^{-1} = M_2^{-1}\left[I - AM_1^{-1}\right] + M_1^{-1},$$

where M_2 is a preconditioner for the second level or stage, whereas M_1 preconditions the first level. In simulations of reservoirs the first level is related to the pressure system. The second level is related to the whole system. The CPR preconditioner satisfies

$$M_{\mathrm{CPR}}^{-1} = M^{-1}\left[I - AC(W^{\mathrm{T}}AC)^{-1}W^{\mathrm{T}}\right] + C(W^{\mathrm{T}}AC)^{-1}W^{\mathrm{T}},$$

where C is an $(n_{\mathrm{eqn}} \cdot n_{\mathrm{cell}})$ by n_{cell} block diagonal matrix (n_{eqn} is the number of unknowns per cell and n_{cell} is the total number of cells in the model). As pressure is the last unknown in each cell, C is given by

$$C = \begin{bmatrix} e_p & & & \\ & e_p & & \\ & & \ddots & \\ & & & e_p \end{bmatrix},$$

$e_p = [0, \ldots, 0, 1]^{\mathrm{T}}$, and W^{T} is an n_{cell} by $(n_{\mathrm{eqn}} \cdot n_{\mathrm{cell}})$ block diagonal matrix. A choice for W^{T} is $W^{\mathrm{T}} = C^{\mathrm{T}}J^{-1}$ where J is a block Jacobi preconditioner. If we see the formula of the CPR preconditioner we remark that a solution of the pressure system is needed in the application of the preconditioner. The authors in Cao *et al.* (2005) propose the use of a multigrid solver, which is efficient in terms of iteration count but lacks scalability for the set-up time. As the application of the preconditioner occurs every iteration we need to solve a linear system of equations corresponding to the pressure matrix (the same in all iterations) every iteration. For the second level we propose the usage of our method EGMRES in the mode of restart and deflation. The procedure of the first level is explained as follows. We follow the notation of the CPR preconditioner and let B be a right-hand side. The system to be solved is $AX = B$. Construct $P(B)$, the enlarged residual, and normalize it (vectors of $P(B)$ are already orthogonal). Let V_1 be the result.

At the first iteration of the second level the application of the preconditioner takes effect on the first block of basis vectors V_1. We are going to explain how to compute

$$M_1^{-1}V_1 = C(W^{\mathrm{T}}AC)^{-1}W^{\mathrm{T}}V_1.$$

Downloaded from https://academic.oup.com/imajna/advance-article-abstract/doi/10.1093/imanum/dry054/5078636
by BIUS Jussieu user
on 28 August 2018

This application is performed by solving the system

$$(W^{\mathrm{T}}AC)X = W^{\mathrm{T}}V_1$$

by using RD-EGMRES and then extending X to the second level by C. Applying W^{T} to V_1 restricts the enlarged residual to the pressure level. Following the definition of the enlarged residual in Section 3, W^TV_1 also has the form of an enlarged residual. Computing

$$(W^{\mathrm{T}}AC)^{-1}W^{\mathrm{T}}V_1$$

is performed as an approximation of the solution. It is obtained by RD-EGMRES. We can write this system in the form

$$(W^{\mathrm{T}}AC)X = W^{\mathrm{T}}V_1.$$

Solving this first level by using RD-EGMRES is natural since as mentioned the right-hand side $W^{\mathrm{T}}V_1$ has the enlarged form. Once we obtain the approximate solution X we extend it to the second level by multiplying it by C. We save in memory the deflation preconditioner M_{def} to use it in the next iterations. Then we continue the rest of the iteration on the second level. To reapply the first-level preconditioner in the next iterations we benefit from the deflation of eigenvalues that we obtained in the first iteration M_{def}. Numerical experiments in Section 6 show that very fast convergence is achieved on both levels.

We note that it is possible to use the EGMRES method only on the pressure level, whereas GMRES (or FGMRES) is used on the saturation level. In application we consider EGMRES to be a flexible preconditioner on the pressure level. Thus, the stopping criterion can be a fixed number of iterations or a large threshold for convergence. The method obtained is more flexible and efficient as the numerical tests presented in Section 6 will show.

6. Numerical experiments

In this section RD-EGMRES stands for EGMRES with restart, deflation of eigenvalues and inexact breakdowns detection. RD-GMRES refers to restarted GMRES with deflation of eigenvalues. Here we investigate the numerical behavior of EGMRES. We compare it to GMRES and BGMRES.

6.1 *Test problems*

Our test matrices arise from the discretization of four types of challenging problems: simulations of reservoirs, seismic imaging, linear elasticity and diffusion problems (Achdou & Nataf, 2007; Niu *et al.*, 2010; Grigori *et al.*, 2014a). All numerical experiments are done using MATLAB 2016R. If it is not otherwise specified the results correspond to RD-EGMRES.

The matrices SKY3D and ANI3D arise from the boundary value problem of the diffusion equation on the three-dimensional unit cube Ω:

$$-\operatorname{div}(\kappa(x)\nabla u) = f \quad \text{in } \Omega, \tag{6.1}$$

$$u = 0 \quad \text{on } \Gamma_D, \tag{6.2}$$

$$\frac{\partial u}{\partial n} = 0 \quad \text{on } \Gamma_N. \tag{6.3}$$

Downloaded from https://academic.oup.com/imajna/advance-article-abstract/doi/10.1093/imanum/dry054/5078636
by BIUS Jussieu user
on 28 August 2018

TABLE 1 *Matrices used for tests. N is the size of the matrix and* NnZ *is the number of nonzero elements. HPD stands for Hermitian positive definite and* κ *is the condition number related to the second norm*

Matrix name	Type	N	NnZ	Real	HPD	κ
BIGCO24	Saturation	752283	5495556	yes	no	2×10^{11}
P-BIGCO24	Pressure	83587	539605	yes	no	10^{9}
BIGP1	Saturation	169328	2469485	yes	no	4×10^{13}
P-BIGP1	Pressure	42332	275946	yes	no	10^{8}
Seismic1	Seismic imaging	11285	55380	no	no	9×10^{3}
Seismic2	Seismic imaging	69611	345450	no	no	6×10^{4}
Seismic3	Seismic imaging	123414	613600	no	no	10^{5}
Elasticity3D100	Elasticity	36663	1231497	yes	yes	3×10^{7}
Elasticity2D125	Elasticity	31752	378000	yes	yes	10^{8}
SKY3D	Skyscraper	8000	53000	yes	yes	10^{5}
ANI3D	Anisotropic layers	8000	53600	yes	yes	10^{3}

The tensor κ is a given coefficient of the partial differential operator, $\Gamma_D = [0,1] \times \{0,1\} \times [0,1]$, Γ_N is chosen as $\Gamma_N = \partial\Omega \setminus \Gamma_D$ and n denotes the exterior normal vector to the boundary of Ω. The matrix ANI3D is obtained by considering anisotropic layers: the domain is made of 10 anisotropic layers with jumps of up to four orders of magnitude and an anisotropy ratio of 10^3 in each layer. Those layers are parallel to $z = 0$, of size 0.1, and inside them the coefficients are constant: $\kappa_y = 10\kappa_x$, $\kappa_z = 100\kappa_x$. This problem is three-dimensional, discretized on a Cartesian grid of size $20 \times 20 \times 20$. The Elasticity3D100 matrix arises from the linear elasticity problem with Dirichlet and Neumann boundary conditions defined as follows:

$$\operatorname{div}(\sigma(u)) + f = 0 \quad \text{in } \Omega, \tag{6.4}$$

$$u = 0 \quad \text{on } \Gamma_D, \tag{6.5}$$

$$\sigma(u) \cdot n = 0 \quad \text{on } \Gamma_N. \tag{6.6}$$

The domain Ω is a unit square (two dimensions or a unit cube three dimensions). The matrices Elasticity3D100 and Elasticity2D125 correspond to this equation discretized using a triangular mesh with $100 \times 10 \times 10$ vertices for the three-dimensional case and 125×10 vertices for the two-dimensional case. Here, Γ_D is the Dirichlet boundary, Γ_N is the Neumann boundary, f is a force and u is the unknown displacement field. The Cauchy stress tensor $\sigma(.)$ is given by Hooke's law: it can be expressed in terms of Young's modulus E and Poisson's ratio ν. n denotes the exterior normal vector to the boundary of Ω. For a more detailed description of the problem see Nataf *et al.* (2013) and Grigori *et al.* (2014b). We consider discontinuous E and ν: $(E_1, \nu_1) = (2 \times 10^{11}, 0.25)$ and discontinuous E in two dimensions: $(E_1, \nu_1) = (10^{12}, 0.45)$ and $(E_2, \nu_2) = (2 \times 10^{6}, 0.45)$. The matrices BIGCO24 and BIGP1 are obtained from our in-house prototype code at Total, which simulates a complex enhanced oil recovery mechanism. This simulator relies on a finite volume discretization and a two-point flux approximation. BIGP1 comes from the simulation of water injection using a black-oil model. The permeability field is heterogeneous (sector model from a real field case). The grid has 42332 active

Downloaded from https://academic.oup.com/imajna/advance-article-abstract/doi/10.1093/imanum/dry054/5078636
by BIUS Jussieu user
on 28 August 2018

TABLE 2 *Comparison between inexact breakdowns detection methods. It_{Method} stands for the number of iterations to achieve convergence. G for GMRES as standard method, EG for EGMRES without inexact breakdowns detection, EG-SVD for EGMRES with inexact breakdowns detection using the SVD, as presented in Calandra et al. (2013), EG-RRQR for EGMRES with inexact breakdowns detection using rank-revealing test (see Section 4.2) and EF is the enlarging factor. Preconditioner:* 128 *block Jacobi. Threshold of convergence is* 10^{-8}

Matrix	It_G	EF	It_{EG}	Dim_{EK}	$It_{\text{EG-SVD}}$	Dim_{EK}	$It_{\text{EG-RRQR}}$	Dim_{EK}
P-BIGCO	319	4	212	848	218	747	218	748
		8	152	1216	158	1124	156	1127
		16	114	1824	125	1622	126	1636
		32	92	2944	98	2313	100	2360
P-BIGP1	822	4	426	1704	435	1601	434	1605
		8	290	2320	294	2176	293	2195
		16	198	3168	201	2929	203	2965
		32	133	4256	139	3856	141	3912
Elasticity3D100	599	4	213	852	213	754	235	779
		8	142	1136	144	945	177	988
		16	101	1616	105	1296	105	1321
		32	78	2496	80	1858	80	1913
Elasticity2D125	1572	4	489	1956	491	1854	490	1858
		8	287	2296	291	2155	289	2168
		16	188	3008	191	2659	190	2682
		32	133	4256	136	3520	136	3566
ANI3D	94	4	85	340	85	314	86	320
		8	80	640	82	589	82	599
		16	75	1200	77	1093	77	1110
		32	65	2080	66	1901	67	1933
SKY3D	376	4	220	880	221	842	233	863
		8	125	1000	125	976	128	981
		16	73	1168	73	1102	73	1109
		32	46	1472	46	1336	46	1347

cells. BIGCO24 corresponds to a simulation of water and gas injection using a compositional model (8 hydrocarbon components). The permeability field is heterogeneous. The grid has 83587 active cells.

In Table 1 we present our test matrices. The linear systems arising from simulations of reservoirs and linear elasticity have one right-hand side. Seismic imaging systems have multiple right-hand sides.

We use a block Jacobi preconditioner with 128 blocks in all our experiments. The threshold of convergence in all our tests is 10^{-8}.

Table 2 shows a brief comparison between EGMRES and GMRES for several matrices in our set. The number of iterations decreases drastically by increasing the enlarging factor of the Krylov subspace. EGMRES and GMRES use the same number of communication messages per iteration. Thus, an overall communication reduction is accomplished by EGMRES. For example, an enlarging factor EF $= 32$ reduces the iteration count by a factor of 12 with the matrix Elasticity2D125. Figures 2 and 3 show that using RRQR or SVD tests to detect the inexact breakdowns does not affect the robustness of the

Downloaded from https://academic.oup.com/imajna/advance-article-abstract/doi/10.1093/imanum/dry054/5078636
by BIUS Jussieu user
on 28 August 2018

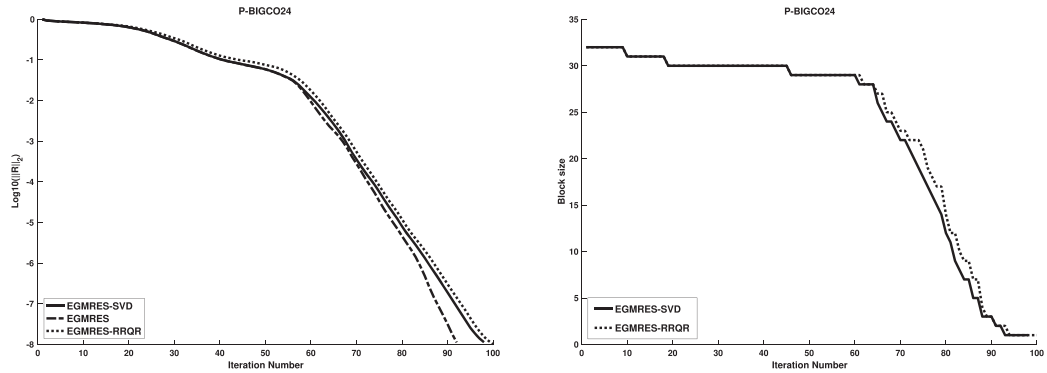

FIG. 2. On the left, the convergence of EGMRES with RRQR and SVD strategies to reduce the size of block vector. On the right, impact of inexact breakdowns detection on the block vector size by using each strategy.

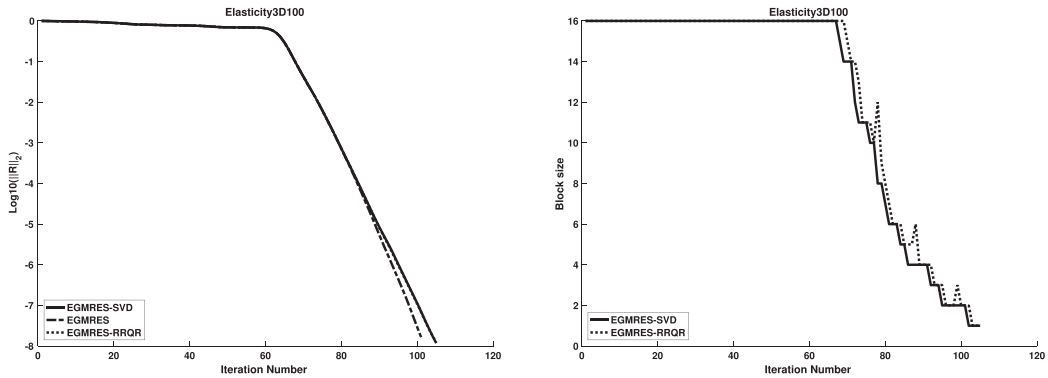

FIG. 3. On the left, the convergence of EGMRES with RRQR and SVD strategies to reduce the size of the block vector. On the right, impact of inexact breakdowns detection on the block vector size by using each strategy.

method. On the contrary they keep the efficiency of the method and they reduce both the memory and the computational costs. A gain in the number of iterations of up to 82% with pressure system P-BIGP1 (EF $= 32$) and with Elasticity3D100 (EF $= 16$) is obtained. We notice also in Fig. 2 that starting from the seventh iteration the size of block vectors that are added to the basis begins to decrease with both strategies, RRQR and SVD. Starting with a block of size 32 EGMRES ends up adding three vectors while maintaining the rate of convergence as if 32 vectors were added at each iteration.

Table 3 outlines the effect of deflating eigenvalues and the accuracy of estimating eigenvectors on the convergence of EGMRES. It shows results for RD-EGMRES(m) with two values of eigenvector convergence threshold. The value of m varies in the set {250, 500}. When the maximal number of vectors to be saved is relatively small, choosing a relatively small EF with a threshold μ (threshold for the criteria of eigenvalues deflation in Algorithm 5) of order 10^{-2} leads to fast convergence and maintains the speed of convergence as if no restart was done. For example, for the challenging matrix P-BIGP1 with EF $= 4$, EGMRES without restart needs 434 iterations to achieve convergence, while RD-EGMRES(250) needs 496 iterations. Comparing to GMRES which iterates 822 times with no restart this difference is small. For our experiments, using a threshold $\mu = 10^{-2}$ is efficient in most cases. Choosing a larger threshold leads to a larger number of eigenvectors being deflated. This yields

Downloaded from https://academic.oup.com/imajna/advance-article-abstract/doi/10.1093/imanum/dry054/5078636
by BIUS Jussieu user
on 28 August 2018

TABLE 3 *Comparison between two tolerance values of residual eigenvectors,* $\mu_1 = 5 \times 10^{-2}$, $\mu_2 = 10^{-2}$. It_{Method} *stands for the number of iterations using* Method *as algorithm, EG(m) for RD-EGMRES where m is the maximum number of stored vectors, either deflated or basis vectors. Preconditioner:* 128 *block Jacobi, '+' means that a stagnation of the norm of the residual occurs and no convergence is achieved. Threshold of convergence is* 10^{-8}

Matrix	EF	It_{EG250}				It_{EG500}			
		μ_1	DimDef	μ_2	DimDef	μ_1	DimDef	μ_2	DimDef
P-BIGCO	1	321	35	322	15	319	0	319	0
	4	222	87	225	34	221	69	220	28
	8	169	145	179	42	161	122	167	43
	16	157	204	182	49	136	193	149	55
	32	+	+	419	55	117	266	160	56
P-BIGP1	1	835	114	835	42	825	72	826	31
	4	512	213	496	81	451	224	450	81
	8	+	+	399	103	331	302	332	105
	16	+	+	1268	137	284	403	277	141
	32	+	+	+	+	+	+	451	166
SKY3D	1	406	22	406	22	376	0	376	0
	4	310	85	307	81	268	86	256	85
	8	193	105	206	96	158	85	159	82
	16	139	119	166	101	102	101	105	96
	32	+	+	+	+	75	102	74	96
ANI3D	1	94	0	94	0	94	0	94	0
	4	86	30	87	1	85	0	85	0
	8	84	52	88	1	82	60	84	1
	16	83	118	100	1	79	103	88	1
	32	108	186	105	2	74	198	89	2
Elasticity2D125	1	1712	224	1689	90	1620	229	1621	85
	4	+	+	650	96	532	256	537	96
	8	+	+	518	109	336	296	356	96
	16	+	+	2245	150	256	357	338	123
	32	+	+	+	+	319	447	394	142
Elasticity3D100	1	608	65	658	29	602	71	632	30
	4	221	82	267	33	216	68	235	28
	8	169	108	220	36	150	104	191	32
	16	159	140	245	40	113	133	160	33
	32	304	186	379	50	99	165	180	38

fewer iterations per cycle, which are not enough to reach convergence fast. A smaller threshold induces deflating a small number of eigenvectors. Thus, we observe a stagnation of the residual. This results in slow convergence. The matrix ANI3D with 128 block Jacobi preconditioner has a small condition number, $\kappa = 232.6$. Thus, the impact of using EGMRES with such system is not important. In

Downloaded from https://academic.oup.com/imajna/advance-article-abstract/doi/10.1093/imanum/dry054/5078636
by BIUS Jussieu user
on 28 August 2018

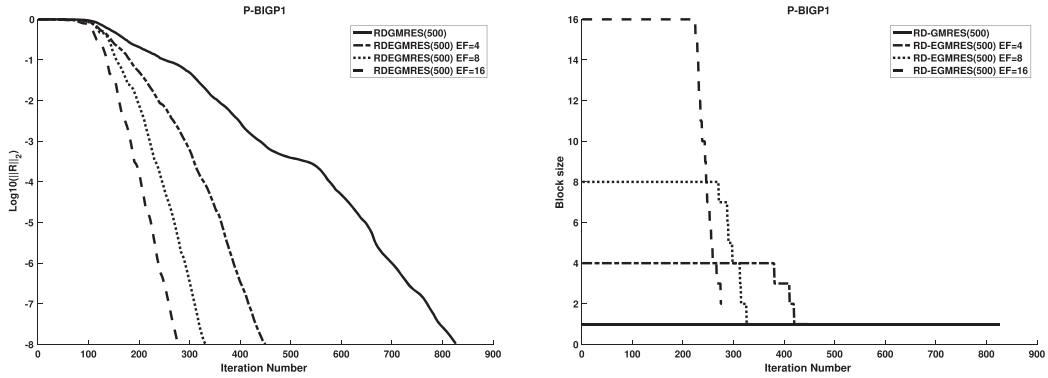

FIG. 4. On the left, RD-EGMRES convergence with different enlarging factors. On the right, impact of inexact breakdowns detection, based on RRQR criterion, on the size of the block vectors.

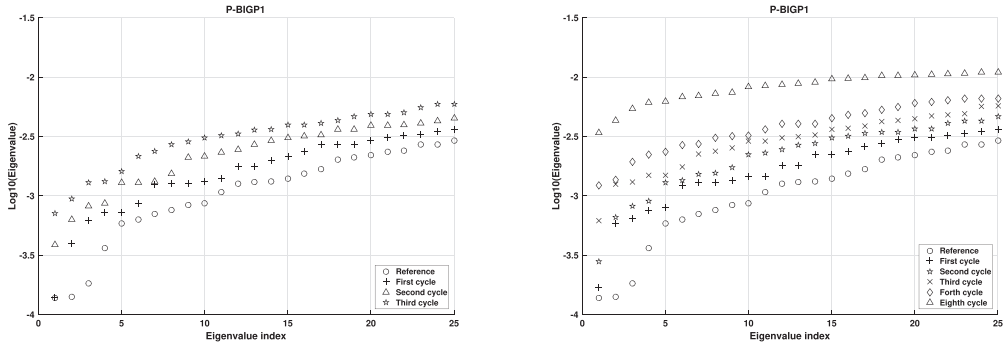

FIG. 5. Eigenvalues deflation for P-BIGP1. On the left EF = 1; on the right EF = 4. Preconditioner: 128 block Jacobi.

most cases, comparing to the full method where no restart is done, RD-EGMRES retains the rate of convergence. For some cases, when the enlarging factor is big and the maximal basis size is small, the method fails to converge. This occurs since the number of iterations per cycle is very few. For example, for the matrix Elasticity2D125 with EF = 32 and a maximal basis size of 250 vectors the method does not converge. Indeed, RD-EGMRES(250) does fewer than seven iterations per cycle. This was not enough to maintain the efficiency. However, with the same basis size and EF = 8 a gain of factor 3.2 is obtained.

6.2 *EGMRES and RD-EGMRES*

Here we present numerical tests for EGMRES and RD-EGMRES on the set of matrices presented in Table 1. In Fig. 4 we show the impact of enlarging the Krylov subspace on the number of iterations to reach convergence. Although the maximal dimension of the search subspace is fixed, increasing the enlarging factor decreases the number of iterations. This efficiency is due to the richness of the enlarged Krylov subspace and the deflation of eigenvalues (see Figs 5 and 6). The number of iterations is reduced by a factor of 3 with EF = 16. Furthermore, we also display the impact of inexact breakdowns detection on the size of the block vectors. Up to EF = 16 RD-EGMRES requires

Downloaded from https://academic.oup.com/imajna/advance-article-abstract/doi/10.1093/imanum/dry054/5078636
by BIUS Jussieu user
on 28 August 2018

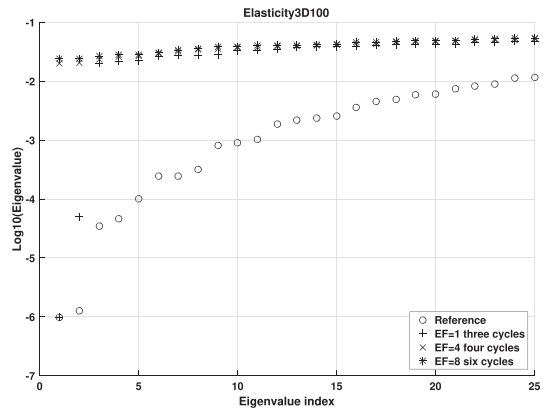

FIG. 6. Comparison: eigenvalues deflation over cycles with different enlarging factors. Eigenvalues deflation on Elasticity3D100. Preconditioner: 128 block Jacobi.

approximately the same number of iterations as the full method (EGMRES) needs to reach convergence. However, the cost of orthogonalization is reduced drastically. Figures 5 and 6 show the efficiency of RD-EGMRES to deflate eigenvalues along cycles. We compute the 25th smallest eigenvalues of the original system as a reference. We run RD-GMRES and RD-EGMRES and compare their ability to deflate eigenvalues. Deflated eigenvalues are shifted such that the spectrum of the deflated system becomes more clustered. In the figure this appears as a translation to the top. We run RD-EGMRES(250) with different enlarging factors EF $=$ 4 or 8. At the end of each cycle we compute the 25th smallest eigenvalues of the deflated system. We do the same for deflated and restarted GMRES with the same basis size 250. We compare with the reference. RD-EGMRES(250) deflates better than the restarted and deflated GMRES. On the right of Fig. 5 RD-EGMRES(250) with EF $=$ 4 deflates eigenvalues after three cycles better than RD-GMRES(250) after three cycles in which it reaches convergence. The first three cycles of RD-EGMRES(250) with EF $=$ 4 perform fewer sparse matrix applications than RD-GMRES(250) performs in the first cycle. In Fig. 6 the test matrix, Elasticity3D100, preconditioned by 128 block Jacobi, has a condition number $\kappa = 2 \times 10^6$. RD-GMRES(250) converges after three cycles without deflating the smallest eigenvalue. RD-EGMRES(250) with EF $=$ 4 and EF $=$ 8 deflates all eigenvalues less than the threshold, chosen as 10^{-2}, after four and six cycles, respectively.

Table 4 shows results for seismic imaging problems with N_p right-hand sides. Enlarging the block Krylov subspace results in faster convergence. In our set of seismic systems we observe that the worse the system is conditioned the more the gain is obtained. Seismic systems 1, 2 and 3 have condition numbers 9×10^3, 6×10^4 and 10^5, respectively. Nevertheless, the gains obtained by RD-EGMRES(500) (comparing to restated and deflated block GMRES) are 45, 53 and 58%, respectively.

6.3 *CPR-EGMRES*

In the following we show results for EGMRES in the context of the CPR preconditioner. Simulation of reservoirs induces linear systems that have a coupling structure of the pressure system, appearing

Downloaded from https://academic.oup.com/imajna/advance-article-abstract/doi/10.1093/imanum/dry054/5078636
by BIUS Jussieu user
on 28 August 2018

TABLE 4 *Comparison between RD-EGMRES and RD-BGMRES. Threshold of convergence is* 10^{-8}. *The maximal size of the search subspace is* 500 *including the deflated eigenvectors*

Matrix	N_p	It_{BGS}	*ND*	It_{EG}
Seismic1	4	468	4	260
			8	326
Seismic2	4	587	4	320
			8	275
Seismic3	4	596	4	309
			8	251

TABLE 5 *Influence of the enlarging factor of the Krylov subspace with multiple right-hand sides, given one at a time.* N_{ev} *is the number of eigenvectors deflated after the solution of* $Ax = B_1$. *Threshold of convergence is* 10^{-8}. *The maximal size of the search subspace is* 500 *including the deflated eigenvectors*

Matrix	EF	$It_{\text{EG}}B_1$	N_{ev}	$It_{\text{EG}}B_2$
P-BIGCO24	1	319	0	319
	4	220	28	125
	8	167	43	100
	16	149	55	94
P-BIGP1	1	826	31	431
	4	450	81	179
	8	332	105	145
	16	277	141	149

as a sub-matrix, and the global system standing for the saturation system. The CPR preconditioning technique (Cao *et al.*, 2005) is widely used for such problems. The main operation while solving the saturation system, by using the CPR preconditioner, is to solve a pressure system at each iteration. For this reason more results on pressure systems are presented rather than other problems. To view the efficiency of using RD-EGMRES to solve the saturation systems, Table 5 presents results for the following type of test: solve the pressure system $AX = B_1$ using RD-EGMRES and save in memory the preconditioner M^{-1} (Algorithm 5) constructed during the solution; then solve the pressure system $AX = B_2$ using RD-EGMRES preconditioned by M^{-1}. In Table 5, to solve $AX = B_2$, RD-GMRES iterates more than RD-EGMRES needs to solve $AX = B_1$. This is very important for CPR preconditioning. Indeed, every application of the CPR preconditioner requires the solution of the first level (Section 5). This yields a linear system of equations with multiple right-hand sides, each given at the application of the preconditioner. Figure 7 illustrates the impact of deflating eigenvalues by using RD-EGMRES. It is true that for the factor EF $= 16$ RD-EGMRES iterates approximately the same number of times as for the factor EF $= 8$. The smallest eigenvalues have been deflated for both cases, such that further improving the conditioning is not useful. However, no stagnation of the residual occurs, in contrast to RD-GMRES. This is related to the comparison between RD-GMRES and RD-EGMRES

Downloaded from https://academic.oup.com/imajna/advance-article-abstract/doi/10.1093/imanum/dry054/5078636
by BIUS Jussieu user
on 28 August 2018

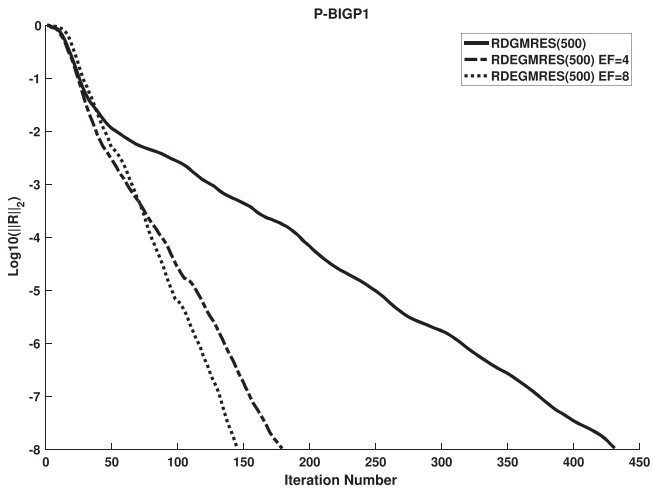

FIG. 7. Deflation of a linear system of equations with multiple right-hand sides, given one at a time. Results show the impact of the enlarging factor on the convergence of $AX = B_2$ and a comparison with RD-GMRES(500). Previously, $AX = B_1$ was solved using RD-GMRES(500) and a deflation preconditioner was constructed to solve $AX = B_2$.

concerning the deflation of the smallest eigenvalues. Two practical approaches for using the CPR-EGMRES preconditioner, precisely on the pressure level, are proposed here:

- use a stopping criterion as the norm of the residual, such that it has to be the same as the one for the saturation system;
- use a specified number of iterations.[1]

For the first type we do not need to use a flexible form while for the second it is necessary to use the flexible variant. Table 6 shows numerical results using these two approaches. For the results of GMRES in Table 6 we use the CPR preconditioner by using a direct *LU* solver in the level of pressure. This is the theoretical CPR approach (Section 5). In our experiments we compare two fixed numbers of iterations, 5 and 10, with the second approach. RD-EGMRES with the CPR preconditioner used for solving the two levels of the system converges faster than the previously described CPR-GMRES solver. Both previously mentioned approaches result in fewer iterations to reach convergence than the CPR-GMRES solver where the subsystem of pressure is solved using a direct *LU* solver. We obtain a gain of up to 50% for the nonflexible variant with EF $= 16$ and up to 35% for the flexible variant with EF $= 16$, when solving systems with BIGCO24 and BIGP1, respectively.

We have to mention that using the standard GMRES, in either the block or simple case, on the saturation level without the flexible variant, causes a stagnation of the real residual norm, whereas the norm of the residual in the (enlarged) Krylov subspace still decreases. This explains why the error in the solution in Table 6 for nonflexible methods is far from the error related to the flexible variant.

[1] The first iteration on the second-level RD-EGMRES performs sufficient iterations on the first level to get deflation information.

Downloaded from https://academic.oup.com/imajna/advance-article-abstract/doi/10.1093/imanum/dry054/5078636
by BIUS Jussieu user
on 28 August 2018

TABLE 6 *CPR-EGMRES. it_G refers to the number of iterations of GMRES by using a direct solver on the pressure level, it_P stands for the fixed number of iterations being done by RD-EGMRES in the pressure level, it_{EG} refers to the number of iterations of EGMRES to reach convergence; it uses RD-EGMRES as a solver for the pressure level. RelErr refers to the relative error in the solution. FEGMRES stands for flexible EGMRES. Threshold of convergence is* 10^{-8}

Matrix	*GMRES*		*EF*	*FEGMRES*			*EGMRES*	
	It_G	RelErr		it_P	it_{FEG}	RelErr	It_{EG}	RelErr
BIGCO24	20	7×10^{-8}	4	5	31	2×10^{-9}	14	8×10^{-6}
			4	10	19	8×10^{-11}		
			8	5	20	2×10^{-9}	12	9×10^{-6}
			8	10	15	3×10^{-10}		
			16	5	17	9×10^{-10}	10	9×10^{-6}
			16	10	14	9×10^{-11}		
BIGP1	81	5×10^{-10}	4	5	78	7×10^{-11}	79	2×10^{-9}
			4	10	68	3×10^{-10}		
			8	5	62	10^{-10}	57	2×10^{-10}
			8	10	59	6×10^{-11}		
			16	5	54	4×10^{-11}	46	2×10^{-9}
			16	10	52	5×10^{-11}		

7. Conclusion

We introduced EGMRES, a linear solver based on two previous works, GMRES (Saad & Schultz, 1986) and, a communication-reducing approach, enlarged Krylov subspace (Grigori *et al.*, 2014a). At each iteration of EGMRES we add multiple basis vectors for each right hand-side, while keeping the same number of messages required for computing this method in parallel. This results in faster convergence. Due to limited memory and the arithmetic cost of orthogonalization, restarting the method is necessary. One solution to this problem is the reduction of the added block vectors, once they are not useful. This maintains the rate of convergence of the method, as if no reduction was done, and decreases the computational and memory costs. Starting from the theory of exact and inexact breakdowns introduced in Robbé & Sadkane (2006) we developed a new theoretical and practical strategy to detect inexact breakdowns based on rank-revealing QR of a $T \times T$ matrix where T is the number of columns of the initial enlarged block residual. This strategy, Algorithm 4, is used to reduce the size of the block vectors. It can be applied to all block GMRES-like methods. The need to solve linear systems of equations with multiple right-hand sides, given one at a time, prompted us to use deflation of eigenvalues to maintain the rate of convergence when a restart occurs. To this end we used Proposition 4.6, originally presented in Erhel *et al.* (1996). This theorem gives an algebraic formulation for a preconditioner once we have the approximate eigenvectors. Thus, we proposed an approach, based on a relative eigenvector residual norm, to choose well-approximated eigenvectors at the end of a restart cycle. This method reduces the number of iterations by a factor up to 7 on our test matrices. We introduced two strategies to use EGMRES as a CPR solver. This solver is used to solve coupled linear systems of equations such as systems arising from simulations of reservoirs. Unlike existing methods, such as proposed in Cao *et al.* (2005), where an algebraic multigrid solves the first level and FGMRES solves the second level,

Downloaded from https://academic.oup.com/imajna/advance-article-abstract/doi/10.1093/imanum/dry054/5078636
by BIUS Jussieu user
on 28 August 2018

EGMRES is used for the two levels of the coupled system and benefits from the deflation of eigenvalues. A gain in the number of iterations of a factor up to 2 is obtained by CPR-EGMRES compared to CPR-GMRES which uses a direct linear solver, *LU*, to solve the pressure level. In conclusion EGMRES reduces the number of iterations to reach convergence. The gain is more important for ill-conditioned linear systems. We compared different thresholds for the criteria of eigenvector approximation. We noticed that a threshold $\varepsilon = 10^{-2}$ leads to good behavior in general. As future work the method will be implemented in parallel. We will also test large linear systems on massively parallel computers and compare the method against multigrid and multilevel domain decomposition methods.

Acknowledgements

The authors sincerely thank the anonymous reviewers and the editor for their constructive suggestions which helped improve the presentation and clarity of this paper.

References

Achdou, Y. & Nataf, F. (2007) Low frequency tangential filtering decomposition. *Numer. Linear Algebra Appl.*, **14**, 129–147.

Agullo, E., Giraud, L. & Jing, Y.-F. (2014) Block GMRES method with inexact breakdowns and deflated restarting. *SIAM J. Matrix Anal. Appl.*, **35**, 1625–1651.

Calandra, H., Gratton, S., Lago, R., Vasseur, X. & Carvalho, L. M. (2013) 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**, S345–S367.

Cao, H., Tchelepi, H. A., Wallis, J. R, & Yardumian, H.E. (2005) SPE Annual Technical Conference and Exhibition, 9-12 October, Dallas, TX: Society of Petroleum Engineers. DOI: https://doi.org/10.2118/96809-MS.

Chan, T. F. (1987) Rank revealing QR factorizations. *Linear Algebra Appl.*, **88**, 67–82.

Chronopoulos, A. & Gear, C. (1989) *s*-step iterative methods for symmetric linear systems. *J. Comput. Appl. Math.*, **25**, 153–168.

El Guennouni, A., Jbilou, K. & Sadok, H. (2003) A block version of BiCGSTAB for linear systems with multiple right-hand sides. *ETNA,* **16**, 129–142 (electronic).

Erhel, J., Burrage, K. & Pohl, B. (1996) Restarted GMRES preconditioned by deflation. *J. Comput. Appl. Math.*, **69**, 303–318.

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

Grigori, L., Moufawad, S. & Nataf, F. (2014a) Enlarged Krylov subspace conjugate gradient methods for reducing communication. *Research Report RR-8597*. Paris, France: INRIA.

Grigori, L., Nataf, F. & Yousef, S. (2014b) Robust algebraic Schur complement preconditioners based on low rank corrections. *Research Report RR-8557*. Paris, France: INRIA.

Gutknecht, M. H. (2007) Block Krylov space methods for linear systems with multiple right-hand sides: an introduction. *Modern Mathematical Models, Methods and Algorithms for Real World Systems* (A. H. Siddiqi, I. S. Duff & O. Christensen eds), pp. 420–447. Anshan.

Gutknecht, M. & Schmelzer, T. (2008) Updating the QR decomposition of block tridiagonal and block Hessenberg matrices generated by block Krylov space methods. *Appl. Numer. Math*., **85**, 871–883.

Hestenes, M. R. & Stiefel, E. (1952) Methods of conjugate gradients for solving linear systems. *J. Res. Natl. Bur. Stan*., **49**, 409–436.

Hoemmen, M. (2010) Communication-avoiding Krylov subspace methods. *Ph.D. Thesis*, Berkeley: EECS Department, University of California.

Karypis, G. & Kumar, V. (1998) Multilevel k-way partitioning scheme for irregular graphs. *J. Parallel Distrib. Comput.*, **48**, 96–129.

Downloaded from https://academic.oup.com/imajna/advance-article-abstract/doi/10.1093/imanum/dry054/5078636
by BIUS Jussieu user
on 28 August 2018

Langou, J. (2003) Iterative methods for solving linear systems with multiple right-hand sides. *Ph.D. Thesis*. Toulouse, France: CERFACS.

Li, T. Y. & Zeng, Z. (2005) A rank-revealing method with updating, downdating, and applications. *SIAM J. Matrix Anal. Appl.*, **26**, 918–946.

Morgan, R. B. (1995) A restarted GMRES method augmented with eigenvectors. *SIAM J. Matrix Anal. Appl.*, **16**, 1154–1171.

Morgan, R. B. (2005) Restarted block-GMRES with deflation of eigenvalues. *Appl. Numer. Math.*, **54**, 222–236.

Nataf, F., Hecht, F., Jolivet, P. & Prud'Homme, C. (2013) Scalable domain decomposition preconditioners for heterogeneous elliptic problems. *SC13*, (Denver, Colorado, USA).

Nikishin, A. A. & Yeremin, A. Y. (1995) 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**, 1135–1153.

Niu, Q., Grigori, L., Kumar, P. & Nataf, F. (2010) Modified tangential frequency filtering decomposition and its Fourier analysis. *Numer. Math.*, **116**, 123–148.

O'Leary, D. P. (1980) The block conjugate gradient algorithm and related methods. *Linear Algebra Appl.*, **29**, 293–322.

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

Saad, Y. (2003) *Iterative Methods for Sparse Linear Systems*, 2nd edn. Philadelphia, PA: Society for Industrial and Applied Mathematics.

Saad, Y. & Schultz, M. H. (1986) GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. Stat. Comput.*, **7**, 856–869.

Simoncini, V. & Gallopoulos, E. (1996) Convergence properties of block GMRES and matrix polynomials. *Linear Algebra Appl.*, **247**, 97–119.

Sonneveld, P. (1989) CGS, a fast Lanczos-type solver for nonsymmetric linear systems. *SIAM J. Sci. Stat. Comput.*, **10**, 36–52.

Stewart, G. W. (1993) Updating a rank-revealing ULV decomposition. *SIAM J. Matrix Anal. Appl.*, **14**, 494–499.

Tang, J. M., Nabben, R., Vuik, C. & Erlangga, Y. A. (2009) Comparison of two-level preconditioners derived from deflation, domain decomposition and multigrid methods. *J. Sci. Comput.*, **39**, 340–370.

Vital, B. (1990) Étude de quelques méthodes de résolution de problèmes lineaires de grande taille sur multiprocesseur. *Ph.D. Thesis*. Rennes, France: University of Rennes 1.

van der Vorst, H. A. (1992) Bi-CGSTAB: a fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems. *SIAM J. Sci. Stat. Comput.*, **13**, 631–644.

Wallis, J., Kendall, R., & Little, T. (1985) Constrained residual acceleration of conjugate residual methods. SPE Reservoir Simulation Symposium, 10-13 February, Dallas, TX: Society of Petroleum Engineers. DOI: https://doi.org/10.2118/13536-MS.

A. Update of QR factorization of the Hessenberg matrix

In this appendix we give a proof of Lemma 4.1. This proof is constructive and explains how we update the QR factorization of the matrix $\tilde{H}$. In Gutknecht & Schmelzer (2008) the authors present strategies to update the factorization of the block Hessenberg matrix. They explain how to update the QR factorization when a different type of deflation is performed.

Proof. of Lemma 4.1 Proof by induction. The case $j = 1$ corresponds to a basic Householder QR factorization. Suppose that the relation holds for j. Let us prove it for $j + 1$. We have in (2.7),

$$\tilde{H}_{j+1} = \begin{pmatrix} H_j & N_{j+1} \\ 0_{s_j,s_j} & M_{j+1} \end{pmatrix}.$$

Downloaded from https://academic.oup.com/imajna/advance-article-abstract/doi/10.1093/imanum/dry054/5078636
by BIUS Jussieu user
on 28 August 2018

Relation (2.9) and the induction hypothesis give the QR factorization of H_j:

$$H_j = \left(\prod_{i=0}^{j-1} \mathscr{Q}^H_{(j+1-i),\,j}\right)\left(\prod_{i=1}^{j} \mathscr{F}_{i,j}\right)\begin{pmatrix} C_j \\ 0_{s,S_j}\end{pmatrix}.$$

We can write

$$\tilde{H}_{j+1} = \left(\prod_{i=0}^{j-1} \mathscr{Q}^H_{(j+1-i),(j+1)}\right)\left(\prod_{i=1}^{j} \mathscr{F}_{i,(j+1)}\right)\begin{pmatrix} \begin{pmatrix} C_j \\ 0_{s,S_j}\end{pmatrix} & \begin{pmatrix} n_{j+1,1} \\ n_{j+1,2}\end{pmatrix} \\ 0_{s_j,S_j} & M_{j+1}\end{pmatrix},$$

where $\begin{pmatrix} n_{j+1,1} \\ n_{j+1,2}\end{pmatrix} = \left(\prod_{i=0}^{j-1} \mathscr{F}^H_{(j-i),\,j}\right)\left(\prod_{i=2}^{j+1} \mathscr{Q}_{(i),\,j}\right) N_{j+1}$.

Let F_{j+1} be the matrix of Householder transformation that triangularizes $\begin{pmatrix} n_{j+1,2} \\ M_{j+1}\end{pmatrix}$; then we obtain the relation satisfied for $j+1$:

$$\tilde{H}_{j+1} = \left(\prod_{i=0}^{j-1} \mathscr{Q}^H_{(j+1-i),(j+1)}\right)\left(\prod_{i=1}^{j+1} \mathscr{F}_{i,(j+1)}\right)\begin{pmatrix} C_{j+1} \\ 0_{s,S_{j+1}}\end{pmatrix}.$$

□

Downloaded from https://academic.oup.com/imajna/advance-article-abstract/doi/10.1093/imanum/dry054/5078636
by BIUS Jussieu user
on 28 August 2018