Enlarged GMRES for reducing communication

Hussam Al Daas, Laura Grigori, Pascal Hénon, Philippe Ricoux

Enlarged GMRES for reducing communication

HUSSAM AL DAAS, AND LAURA GRIGORI
Inria Paris, Alpines, and UPMC Univ Paris 06, CNRS UMR 7598, Laboratoire Jacques-Louis Lions.

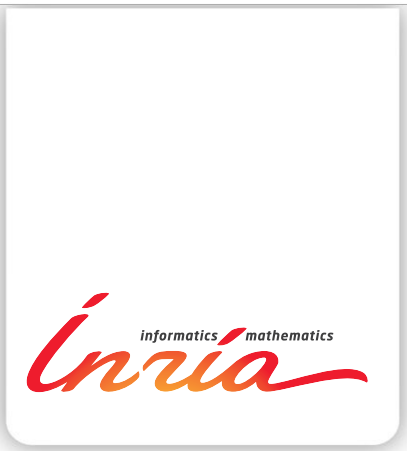
PASCAL HÉNON
TOTAL EP, Centre Scientifique et Technique Jean Féger, Pau, France.

AND

PHILIPPE RICOUX
TOTAL SA, R&D Group, Tour Michelet La défense 10, Paris-La Defense, France.

RESEARCH REPORT
N° 9049
March 2017
Project-Team Alpines

ISSN 0249-6399 ISRN INRIA/RR--9049--FR+ENG



Enlarged GMRES for reducing communication

Hussam AL DAAS*, and Laura GRIGORI†
Inria Paris, Alpines, and UPMC Univ Paris 06, CNRS UMR 7598, Laboratoire Jacques-Louis Lions.

Pascal Hénon
TOTAL EP, Centre Scientifique et Technique Jean Féger, Pau, France.

and

Philippe RICOUX
TOTAL SA, R&D Group, Tour Michelet La défense 10, Paris-La Defense, France.

Project-Team Alpines

Research Report n° 9049 — March 2017 — 32 pages

Abstract: We propose a variant of the 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 an eigenvalues deflation technique aiming to have 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 a very fast convergence when solving the same system with different right-hand sides, each given at a different time (useful in the context of CPR preconditioner). With the same memory cost, we obtain a saving of up to 50% in the number of iterations to reach convergence with respect to the original method.

Key-words: Krylov iterative methods, linear solvers, multiple right-hand sides, inexact breakdowns, deflation of eigenvalues, communication avoiding

* Corresponding author. Email: hussam.al-daas@inria.fr
† Email: laura.grigori@inria.fr

RESEARCH CENTRE
PARIS – ROCQUENCOURT

Domaine de Voluceau, - Rocquencourt
B.P. 105 - 78153 Le Chesnay Cedex

Enlarged GMRES for reducing communication

Résumé : Nous proposons une variante de la méthode GMRES pour résoudre des systèmes linéaires. Cette variation est basée sur l'idée du sous-espace de Krylov élargi afin de réduire les communications. Elle peut être considérée comme un cas particulier de GMRES par bloc. Nous nous sommes intéressés à la détection des inexact breakdowns. Nous proposons une nouvelle stratégie qui permet cette détection. En outre, nous proposons une technique pour deflater les valeurs propres avec deux avantages. Premièrement, elle évite le plateau de convergence après la fin du cycle lorsqu'on en recommence un nouveau. Deuxièmement, elle converge rapidement quand il faut résoudre le même système avec différents second-membres, donnés les uns à la suite des autres. Avec le même coût mémoire, nous obtenons une réduction de 50% du nombre d'itérations pour atteindre la convergence comparé à la méthode originale.

Mots-clés : Méthodes de Krylov, solveurs linéaire, méthodes par bloc, inexact breakdowns, déflation de valeurs propres, communication avoiding

Enlarged GMRES for reducing communication

March 29, 2017

1 Introduction

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

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

where $X \in \mathbb{C}^{n\times s}$, and $B \in \mathbb{C}^{n\times s}$ is full rank, with $s \geq 1$ 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 systems of equations. They find a sequence of approximate solutions $X_1, \ldots, X_j$ respectively in the affine spaces $X_0 + \mathcal{K}_j(A, R_0)$, where X_0 is the initial guess, R_0 is the corresponding initial residual and

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

is the j^{th} block Krylov subspace related to A and R_0.

Generalized Minimal RESidual (GMRES) [24], Conjugate Gradient (CG) (Hermitian case) [13], Conjugate Gradient Squared (CGS) [27] and Bi-Conjugate Gradient STABilized (BiCGStab) [29] 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 since the computation part becomes negligible when the number of processors increases. Thus, the block-type methods were introduced. These schemes have three main advantages. Firstly, matrix-set of vectors operations are used. Secondly, the solution of multiple right-hand-sides are computed simultaneously. Lastly, a faster convergence can be achieved by using a larger search subspace. Generally, simple Krylov subspace methods have a block variant, (e.g. block GMRES [30], block BiCGStab [6], block CG [21]). 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 [23]. O'Leary [21], studies the convergence analysis of block conjugate gradient and presents an estimation of the error in the approximate solutions. In [26], Simoncini and Gallopoulos generalize the theory of convergence presented in [21] to the block GMRES method. This generalization is restricted to the special case when the real part of the spectrum is positive definite.

Hybrid methods also exist, as s-step methods [5,14] which are based on the idea of performing s iterations of the simple method in one iteration. Recently, the enlarged Krylov subspace

*Corresponding author. Email: hussam.al-daas@inria.fr
†Email: laura.grigori@inria.fr

approach was introduced in [9] along with a communication reducing conjugate gradient based on it.

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 [8, 15]. Different strategies to deal with this issue are presented in the literature [2, 3, 8, 12, 15, 19, 22]. To detect inexact breakdowns in block-like GMRES a rank test has to be done at each iteration. In [2, 3, 22], the authors propose an inexact breakdowns detection test based on SVD factorization of the block residual in the block Krylov subspace. This strategy implies the solution of the least squares problem at each iteration in order to obtain the block residual in the block Krylov subspace, then it performs its SVD factorization. The dimension of the block Krylov residual increases linearly with the number of iteration.

Solving challenging and large scale linear system of equations by a long recurrence Krylov method requires restarting the method which leads to a slow convergence. To avoid this issue it is common to use the deflation of eigenvalues [7, 16, 17]. Before restarting (block) GMRES, either ritz values or harmonic ritz values are computed to be used by deflation.

In this paper we focus on the GMRES scheme as presented in [23]. We introduce Enlarged GMRES method, which is based on the enlarged Krylov subspace methods [9]. It is adapted for solving linear systems of equations with one or multiple right-hand sides.

The paper is organized as follows. In section 2, we give a brief discussion about existing variants of GMRES and its block version. We review exact and inexact breakdowns as introduced in [22]. We review the deflated Arnoldi procedure and the inexact breakdowns detection test that is proposed in [22].

In section 3, we introduce EGMRES. It adds multiple basis vectors at each iteration. In section 4 we give a new strategy to reduce the size of the block basis vectors. We prove that the inexact breakdowns detection can be performed on a small matrix of size $s \times s$ rather than a size of approximately $js \times s$, where s is the number of columns in the initial block residual and j is the iteration number. In addition, we show how this $s \times s$ matrix is computed iteratively, while in [3, 22] it is necessary to solve a least squares problem and compute a representation of the block residual in order to perform the rank deficiency test. Furthermore, we study a new strategy based on rank-revealing QR to reduce the size of the block in 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 [22], and we show that they have approximately the same behavior.

We show experimentally that the enlarged Krylov subspace method approximates better the eigenvalues of the input matrix than the classical GMRES method for a same size of the basis. This basis is built with a smaller number of iterations for the enlarged Krylov subspace method and hence less communication cost. 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 Enlarged GMRES or RD-EGMRES. By using RD-EGMRES, we obtain a gain of a factor of up to 7 with respect to GMRES in terms of number of iterations on our set of matrices.

Section 5 presents CPR-EGMRES, a special linear solver for coupled systems. Since we are interested in solving linear systems arising from reservoir simulations, we adapt EGMRES to be used as a CPR solver [31]. Such linear systems are formed by two coupled systems. Unlike the common choice of algebraic multigrid, proposed in [25] to solve the first level of the coupled system, we propose two practical strategies that use EGMRES on both levels. The first level sub-system is solved at each iteration. Thus, we benefit from the approximation of eigenvalues provided by the enlarged GMRES method and their deflation. The difference between the two

strategies that we propose lies in the first level. The first strategy uses a fixed number of iterations without the necessity to reach 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 second level. Note that the second strategy can be considered as a linear operator by reason of convergence, 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 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, each given at a time. This is related to the CPR preconditioner that is 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 Notations

Matlab notations are used in a block sense: $M(i,j)$ is the element in the block line i and the block column j of the block matrix M. $(M(i,j))_{i,j}$ represents a block matrix M whose block elements are $M(i,j)$. $\|.\|_F$ represents the Frobenius norm. Let $t > 0$ 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 $s_j \leq s$ be the number of added vectors to the basis of the block Krylov subspace $\mathcal{K}_{j-1}(A, R_0)$ at iteration j, and $c_j = s - s_j$. $S_j = \sum_{i=1}^{j} s_i$ is the dimension of the block Krylov subspace $\mathcal{K}_j(A, R_0)$. 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. V^H represents the conjugate of V. $V^\top$ 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 X_j and X_j^E the solution and the enlarged solution. $\oplus$ refers to the direct sum between orthogonal subspaces. Finally, we define the following notations: $\tilde{V}_{j+1} \in \mathbb{C}^{n \times s_j}$ 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 $\mathcal{K}_j(A, R_0)$, to get $\mathcal{K}_{j+1}(A, R_0)$. $\mathcal{V}_j = [V_1, \cdots, V_j] \in \mathbb{C}^{n \times S_j}$ denotes the matrix whose columns are the basis vectors of the block Krylov subspace $\mathcal{K}_j(A, R_0)$. $D_j \in \mathbb{C}^{n \times c_{j+1}}$ 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. $\mathcal{V}_{m+1}$, 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: $\mathcal{V}_m = [V_1, \cdots, V_m]$, $\mathcal{V}_{m+1} = [\mathcal{V}_m, V_{m+1}]$, $H_m = (H(i, j))_{i,j}$.
9: **end for**

The block generalized minimal residual method (see Algorithm 2), BGMRES [30], is a Krylov subspace method. It finds a sequence of approximate solutions X_j, $j > 0$, for the system of linear equations $AX = B$. The residual norm $\|R_j\|_F$ is minimal over the corresponding block Krylov subspace

$$\mathcal{K}_j(A, R_0) = BlockSpan\{R_0, AR_0, \ldots, A^{j-1}R_0\}. \tag{2}$$

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 $\mathcal{V}_{m+1}$ and H_m using Block-Arnoldi (A, V_1, m) (Algorithm 1).
4: Solve the least squares problem $Y_m = \underset{Y\in\mathbb{C}^{ms\times s}}{\arg\min} \|H_m Y - E_1\Pi_0\|_2$,
where $E_1 = (I_s, 0_{m,s})^\top \in \mathbb{C}^{js\times s}$.
5: $X_m = X_0 + \mathcal{V}_m Y_m$.

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

$$A\mathcal{V}_j = \mathcal{V}_j H_j(1 : j, 1 : j) + V_{j+1}H_j(j + 1, j)E_j^\top, \tag{3}$$

where $\mathcal{V}_j = [V_1, \cdots, V_j]$, and $E_j = (0_{s,(j-1)s}, I_s)^\top \in \mathbb{C}^{js\times s}$. A detailed overview of block Krylov methods is given in [12].

2.3 Block Arnoldi and exact breakdown

Definition 1. *A subspace $\mathcal{S} \subset \mathbb{C}^n$ is called A-**stable** if it is invariant under the multiplication by A, i.e. $\forall u \in S$, $Au \in S$.*

The importance of having an A-stable 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-stable subspace is contained in the Krylov subspace.

Definition 2. *An exact breakdown [22] is a phenomenon that occurs at the j^{th} 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 $c_{j+1} = s - rank(W)$ where s is the rank of V_1.*

The following lemma [23] illustrates the importance of the breakdown in GMRES.

Lemma 1. *In GMRES, when a breakdown occurs during an iteration j, the Krylov subspace*

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

is an A-stable subspace.

Proof. A breakdown in GMRES occurs during the iteration j when $H_j(j+1, j) = 0$. Thus, immediately from relation (3), we get $A\mathcal{V}_j = \mathcal{V}_j H_j(1 : j, 1 : j)$. It yields that the subspace $Span\{V_1, \cdots, V_j\}$ is A-stable. □

However, in general, for the block Arnoldi procedure, an exact breakdown does not mean that there is an A-stable 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-stable. We recall several equivalent conditions related to the exact breakdown in Theorem 1. For the details and the proof see [22]. Let c_{j+1} denote the rank deficiency of W (Line 6, Algorithm 1), i.e. $c_{j+1} = s - rank(W)$, where s is the rank of V_1.

Theorem 1. *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. *$dim\{Range(V_1) \cap A\mathcal{K}_j(A, R_0)\} = c_{j+1}$.*
3. *$rank(R_j) = s - c_{j+1}$.*
4. *$dim\{Range(X) \cap \mathcal{K}_j(A, R_0)\} = c_{j+1}$.*

Proof. See the proof in [22]. □

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 Block GMRES method with deflation at each iteration (referred to as IBBGMRES-R) proposed by Robbé and Sadkane [22]. This method was then reformulated in a different way by Calandra et al. [3] (referred to as BFGMRES-S).

2.3.1 Deflated Arnoldi relation

Here, we review the derivation of the modified algebraic relations of the Arnoldi procedure, presented in e.g. [3, 22]. We follow the presentation in [3]. 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 $\mathcal{V}_j$ 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

$$Range([\tilde{V}_{j+1}, D_{j-1}]) = Range(V_{j+1}) \oplus Range(D_j), \tag{4}$$

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

$$A\mathcal{V}_j = [\mathcal{V}_{j+1}, D_j]H_j, \tag{5}$$

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

$$A\mathcal{V}_{j-1} = [\mathcal{V}_j, D_{j-1}]H_{j-1}. \tag{6}$$

Let us study the iteration j. First, we multiply A by V_j. Then, we orthogonalize against $\mathcal{V}_j$ 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,

$$A\mathcal{V}_j = [\mathcal{V}_j, D_{j-1}, \tilde{V}_{j+1}]\tilde{H}_j. \tag{7}$$

$\tilde{H}_j$ has the form

$$\tilde{H}_j = \begin{pmatrix} H_{j-1} & N_j \\ 0_{s_j, S_{j-1}} & M_j \end{pmatrix} \tag{8}$$

where $N_j = [\mathcal{V}_j, D_{j-1}]^H AV_j \in \mathbb{C}^{(S_{j-1}+s)\times s_j}$ and $(AV_j - [\mathcal{V}_j, D_{j-1}]N_j) = \tilde{V}_{j+1}M_j$ is the QR factorization. To transform the relation (7) to the form in (5), let $Q_{j+1} \in \mathbb{C}^{s\times s}$ be a unitary matrix such that

$$[D_{j-1}, \tilde{V}_{j+1}]Q_{j+1} = [V_{j+1}, D_j], \tag{9}$$

then, we have

$$A\mathcal{V}_j = [\mathcal{V}_{j+1}, \mathcal{D}_j]\mathcal{Q}_{j+1}^H \tilde{H}_j, \tag{10}$$

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

$$A\mathcal{V}_j = [\mathcal{V}_{j+1}, D_j]H_j. \tag{11}$$

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 remaining of this section, we show the inexact breakdown detection as presented in [3, 22].

2.4 Inexact breakdown and subspace decomposition

In [22], the authors introduce exact and inexact breakdown in the BGMRES-like methods. They define the inexact breakdown as the following.

Definition 3. *An inexact breakdown is a phenomenon that occurs when the matrix* $\begin{pmatrix} R_0 & AR_0 & \dots & A^m R_0 \end{pmatrix}$ *becomes almost rank deficient.*

Detecting inexact breakdowns and deflating useless vectors leads to less computation and more memory for useful vectors. In [22], 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 [3]. We start from relation (9). 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 subspace that has not converged of the block residual. Let $\bar{R}_j = U\Sigma W^H$ be the SVD factorization of the block Krylov residual. In [22], the authors decompose this factorization as

$$\begin{aligned}\bar{R}_j &= \begin{pmatrix} U_1 & U_2 \\ U_{s+} & U_{s-} \end{pmatrix} \begin{pmatrix} \Sigma_1 & \\ & \Sigma_2 \end{pmatrix} [W_1, W_2]^H \\ &= \begin{pmatrix} U_1 \\ U_{s+} \end{pmatrix} \Sigma_1 W_1^H + \begin{pmatrix} U_2 \\ U_{s-} \end{pmatrix} \Sigma_2 W_2^H, \end{aligned} \tag{12}$$

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 $\mathcal{K}_{j+1}$ is given by

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

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

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

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 Section 4.1, we propose a new strategy to avoid this costly test by reducing the dimension of the problem. In addition, a study of inexact breakdowns detection based on rank revealing QR is given.

3 Enlarged GMRES

We introduce in this section our new block GMRES method EGMRES. This new scheme is based on the enlargement of the block Krylov subspace [9]. 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. This method depends on the partition of the set of unknowns.

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

$$P_i : \mathbb{C}^{n\times s} \to \mathbb{C}^{n\times s} \tag{13}$$

$$u \to Z_i Z_i^H u, \tag{14}$$

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

$$u = \sum_{i=1}^{t} P_i(u), \quad \forall u \in \mathbb{C}^{n\times s}; \tag{15}$$

$$P_i \perp P_j, \quad i \neq j. \tag{16}$$

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

$$P(R_0) = [P_1(R_0), \cdots, P_t(R_0)]. \tag{17}$$

Definition 4. *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 j^{th} t-enlarged Krylov subspace associated with the matrix A and the initial residual R_0 is defined by*

$$\mathcal{K}_{j,t}(A, R_0) = BlockSpan\{P(R_0), AP(R_0), \cdots, A^{j-1}P(R_0)\}, \tag{18}$$

where the projection P is given by (16).

We will refer to the enlarged Krylov subspace $\mathcal{K}_{j,t}(A, R_0)$ by $\mathcal{K}_{j,t}$ when there is no ambiguity. For more details on the enlarged Krylov subspaces we refer the reader to [9].

Definition 5 (Enlarged GMRES)**.** *The Enlarged GMRES, denoted EGMRES, is an enlarged Krylov subspace method. It finds a sequence of approximate solutions $\{X_1, \cdots, X_m\}$ for the system of linear equations $AX = B$. $X_j - X_0$ belongs to the j^{th} enlarged Krylov subspace $\mathcal{K}_{j,t}(A, R_0)$ with R_0 the initial residual. $\|R_j\|_F = \|B - AX_j\|_F$ is minimal over the enlarged Krylov subspace.*

3.1 Enlarged GMRES algorithm

The following algorithm is the basic form of Enlarged GMRES. Let $\mathbf{1}_t = [I_s, \cdots, I_s]^\top \in \mathbb{C}^{T\times s}$ with I_s the identity matrix of size s.

The update of the Hessenberg matrix (Line 8, Algorithm 3) means updating its QR factors $\mathcal{F}_j$ and C_j such that $H_j = \mathcal{F}_j \begin{pmatrix} C_j \\ 0_{T,jT} \end{pmatrix}$, where $\mathcal{F}_j \in \mathbb{C}^{(j+1)T\times(j+1)T}$ 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 $\mathcal{K}_{j,t}(A, R_0)$.

Proposition 1. *Following the notations in algorithm 3 we have*

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

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 (17).
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 $H_j = \mathcal{F}_j \begin{pmatrix} C_j \\ 0_{T,jT} \end{pmatrix}$
where $\mathcal{F}_j \in \mathbb{C}^{(j+1)T\times(j+1)T}$ and $C_j \in \mathbb{C}^{jT\times jT}$.
9: Compute $\begin{pmatrix} E_j \\ G_j \end{pmatrix} = \mathcal{F}_j^H \Pi_j$
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 $Y_j = \underset{Y\in\mathbb{C}^{jT\times T}}{\arg\min} \|\Pi_j - H_jY\|$,
$Y_j = C_j \setminus E_j$.
15: $X_j = X_0 + [V_1, \cdots, V_j]Y_j\mathbf{1}_t$.

Proof. We have

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

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

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

□

After the presentation of the EGMRES method, we remark that once we enlarge the block residual, it returns to block GMRES scheme. The difference is represented by recovering the solution of each right-hand side. It is done by summing the solution vectors related to each right-hand side.

4 Inexact breakdowns and eigenvalues deflation in block GMRES

As mentioned previously, EGMRES has a block GMRES scheme. For this, in this section we study block GMRES rather than EGMRES. The application of this study on EGMRES is natural.

Relations in Lemma 2 and Proposition 2 hold untill the end of this paper.

4.1 Inexact breakdowns

In this section we give a new strategy to detect inexact breakdowns. In [3,22], 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 $\bar{R}_j \in \mathbb{C}^{(S_j+s)\times s}$. This matrix has a dimension that depends on the iteration number j. Proposition 2 is the key idea to reduce the dimension of the SVD problem. Before that we need the following lemma 2. This lemma is going to be a tool in the remaining of this section.

Lemma 2. *The QR factorization of the matrix $\tilde{H}_j$ in the relation (7) is given by the relation*

$$\tilde{H}_j = \left(\prod_{i=0}^{j-2} \mathcal{Q}_{(j-i),j}^H\right)\left(\prod_{i=1}^{j} \mathcal{F}_{i,j}\right)\begin{pmatrix} C_j \\ 0_{s,S_j}\end{pmatrix}, \tag{19}$$

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.* $Q_i \in \mathbb{C}^{s\times s}$ *is the rotation matrix obtained by the inexact breakdowns test.* $F_i \in \mathbb{C}^{(s_i+s)\times(s_i+s)}$ *is the Householder transformation matrix used to triangularize the block* $\tilde{H}_i(i : i+1, i)$ *after updating* $\tilde{H}_i(1 : i, i)$ (7) *by using* F_k *for* $k = 1, \cdots, i-1$. *With the convention* $S_0 = 0$.

Proof. Proof is constructive and is given in the Appendix. □

Proposition 2. *The following relations hold during the block GMRES with 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. $\|B - A(X_0 + \mathcal{V}_j Y)\|_F = \| \left(\prod_{i=0}^{j-1} \mathcal{F}_{(j-i),j}^H\right) \mathcal{Q}_{1,j}^H \begin{pmatrix} \Pi_0 \\ 0_{S_j,s} \end{pmatrix} - \begin{pmatrix} C_j \\ 0_{s,S_j} \end{pmatrix} Y \|_F$.
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* $G_j \in \mathbb{C}^{s \times s}$.

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 deflation technique based on detection of inexact breakdowns.

As explained in [22], Robbé and Sadkane introduced two criteria based on singular value decomposition to determine the convergent subspace. The first depends on the block residual. The second depends on the block basis vector. In a later paper [3], Calandra et al. 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 breakdown detection

Here we present how we reduce 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 for all block GMRES-like methods.

Proposition 3. *An SVD factorization on the matrix* G_j *is equivalent to an SVD factorization of* $\bar{R}_j$.

Proof. Proposition 2 proves that

$$\bar{R}_j = \left(\prod_{i=0}^{j-2} \mathcal{Q}_{(j-i),j}^H\right) \left(\prod_{i=1}^{j} \mathcal{F}_{i,j}\right) \begin{pmatrix} 0_{S_j,s} \\ G_j \end{pmatrix} \tag{20}$$

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

Proof. Proof is similar to the proof of Proposition 3 □

An important difference related to the reference test presented in [3,22] is that the dimension of the factorized matrix does not depend on the iteration number j. In the mentioned references this dimension is $S_j \times s$ at iteration j. Proposition 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 remaining of this section we introduce an inexact breakdown detection based on rank revealing QR to reduce the cost of performing an SVD factorization.

We start from relation (9). 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 $\begin{pmatrix}0_{S_j,s}\\ G_j\end{pmatrix} = \left(\prod_{i=0}^{j-1} \mathcal{F}_{(j-i),j}^H\right)\left(\prod_{i=2}^{j} \mathcal{Q}_{i,j}\right)\bar{R}_j$ as presented in Proposition 2, find Q_{j+1} such that V_{j+1} spans the subspace related to the non convergent part of the block residual.

In [3] and [22], the authors propose a strategy based on the singular value decomposition 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 equals to $s \times s$. Furthermore, using rank revealing QR factorization [4] instead of SVD factorization reduces the computational complexity. Here, we derive the theory of that strategy.

Let ε_0 be a threshold given, and $G_j = S\mathcal{R}P^\top$ 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, $\mathcal{R}$ is an upper triangular matrix and P is a permutation matrix. We can write the rank revealing QR relation in the form,

$$\begin{aligned} G_j &= \begin{pmatrix}S_+ & S_-\end{pmatrix}\begin{pmatrix}\mathcal{R}_1 & \mathcal{R}_2\\ 0_{c_{j+1},s_{j+1}} & \mathcal{R}_3\end{pmatrix}\begin{pmatrix}P_1^\top\\ P_2^\top\end{pmatrix}\\ &= S_+\begin{pmatrix}\mathcal{R}_1 & \mathcal{R}_2\end{pmatrix}\begin{pmatrix}P_1^\top\\ P_2^\top\end{pmatrix} + S_-\mathcal{R}_3P_2^\top, \end{aligned} \tag{21}$$

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

To detect inexact breakdowns by using RRQR, the test depends on the same idea that is proposed in [3, 22]. The new basis vectors to be added should be related to the subspace that has not converged of the range of $\bar{R}_j$. We write the projection of the residual on the subspace perpendicular to $\mathcal{K}_j$ using the RRQR decomposition,

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

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

$$(I - \mathcal{V}_j \mathcal{V}_j^H) R_j = [0, D_{j-1}, \tilde{V}_{j+1}] \bar{R}_j \tag{22}$$

$$= [0, D_{j-1}, \tilde{V}_{j+1}] \left(\prod_{i=0}^{j-2} \mathcal{Q}_{(j-i),j)}^H \right) \left(\prod_{i=1}^{j} \mathcal{F}_{i,j} \right) \begin{pmatrix} 0_{S_j,s} \\ S \end{pmatrix} \mathcal{R} P^\top \tag{23}$$

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

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

$$= Range\left([D_{j-1}, \tilde{V}_{j+1}] \underline{S} \right). \tag{24}$$

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

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

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 new 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 show how to compute the matrix Q_{j+1} (10).

Algorithm 4 Inexact breakdown 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, $G_j = S\mathcal{R}P^\top$.
2: $G_j = S_+(\mathcal{R}_1 P_1^\top + \mathcal{R}_2 P_2^\top) + S_- \mathcal{R}_3 P_2^\top$. With $\mathcal{R}_3$ has maximum size with second norm less than ε. s_{j+1} is the rank of G_j.
3: QR factorization of $\underline{S}$ (24), 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 subspace, and rebuilding a new one beginning with the last residual. This means a loss of information. A common approach to keep useful information is to deflate small eigenvalues if their eigenvectors have converged or if they are well approximated by the end of the cycle [7, 10, 16, 28].

In the remaining of this section, we show how these eigenvalues and eigenvectors are chosen. We first recall a theorem from [7]. 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 2. *Suppose that the matrix A is diagonalizable and let $\{\lambda_1, \cdots, \lambda_n\}$ be the set of eigenvalues of A with $|\lambda_1| \leq \cdots \leq |\lambda_n|$ and $\{u_1, \cdots, u_n\}$ be the corresponding normal eigenvectors. Given a threshold ε_1 set m to be the positive integer such that $|\lambda_m| < \varepsilon_1 \leq |\lambda_{m+1}|$. Let $U = (u_1, \cdots, u_m) = ZL$ be the QR factorization and $M = I_n + Z(\frac{1}{|\lambda_n|} Z^H A Z - I_m) Z^H$, then*

1. *The matrix M is inversible and its inverse $\tilde{M} = I_n + Z(|\lambda_n|(Z^HAZ)^{-1} - I_m)Z^H$.*
2. *The matrix AM^{-1} has eigenvalues $\{\lambda_n, \cdots, \lambda_n, \lambda_{m+1}, \cdots, \lambda_n\}$ with $\{u_1, \cdots, u_n\}$ as corresponding eigenvectors.*

Proof. A similar proof using invariant subspaces is given in [7]. □

4.3.1 Well approximated eigenvalues and eigenvectors

Let S_1 be the upper $k_{cycle} \times k_{cycle}$ sub-matrix of the matrix H_{cycle}, where *cycle* is the number of last iteration in block GMRES. Suppose that S_1 is diagonalizable (it is sufficient to suppose that so is A), and let $\{\lambda_1, \cdots, \lambda_m\}$ be the eigenvalues of S_1 with absolute value less than a threshold, ε_1, given. 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. *Let H_{cycle} be the matrix verifying*

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

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

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

Proof.

$$\begin{aligned} A\mathcal{V}_{cycle}u &= [\mathcal{V}_{cycle+1}, D_{cycle}]H_{cycle}u \\ &= [\mathcal{V}_{cycle+1}, D_{cycle}] \begin{pmatrix} S_1 \\ S_2 \end{pmatrix} u \\ &= \mathcal{V}_{cycle}S_1u + [V_{cycle+1}, D_{cycle}]S_2u \\ &= \lambda\mathcal{V}_{cycle}u + [V_{cycle+1}, D_{cycle}]S_2u. \end{aligned}$$

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

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

□

Scaling the spectrum of the linear system is taken into account in practice. To decide if a vector $\mathcal{V}_{cycle-1}u$ approximates well 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 such eigenvalue converges fast.

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.

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

Require: The matrix A, the basis of the block Krylov subspace $\mathcal{V}_{cycle}$, 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 $Z^H AZ$, an approximation of the magnitude of the largest eigenvalue $|\lambda_{max}|$ (optional).

1: Calculate the eigenvalues $\{\lambda_1, \cdots, \lambda_p\}$ of S_1 with absolute value less than ε and their corresponding normal eigenvectors $\{u_1, \cdots, u_p\}$.
2: Set $U = [\,]$.
3: **if** $|\lambda_{max}|$ is not provided **then**
4: Compute $|\lambda_{max}|$.
5: **end if**
6: **for** $i = 1 : min\{p, nev\}$ **do**
7: **if** $\|S_2 u_i\|_2 < |\lambda_{max}|\varepsilon$ **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 $\mathbb{C}^n$, $Z = \mathcal{V}_{cycle} Z$.
14: Form $Z^H AZ = L^H \Lambda L$ to use in the preconditioner, Λ is a diagonal matrix with the deflated λ_i.

5 CPR-EGMRES

In this section, we introduce the constrained pressure residual preconditioner with EGMRES. This preconditioner was first introduced by [31] as a preconditioner for the solution of system of linear equations coming from the simulation of reservoirs. In reservoir simulations, 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 [25, 31]. As a direct consequence, the linear systems in reservoir simulations are a natural target for a two-stage preconditioning strategy.

5.1 Two-stage preconditioning

The two-stage preconditioning formula is given by

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

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

$$M_{CPR}^{-1} = M^{-1}[I - AC(W^T AC)^{-1}W^T] + C(W^T AC)^{-1}W^T.$$

where C is an $(n_{eqn} \cdot n_{cell})$ 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

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 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 $nev = nev_{max}$, $R_0 = B - AX_0$.
for $cycle = 1 : cycle_{max}$ **do**
  if $cycle > 1$ and $nev > 0$ **then**
    Call the 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 (19).
    $$\begin{pmatrix} E_j \\ G_j \end{pmatrix} = \mathcal{F}_{j,j}^H \begin{pmatrix} E_{j-1} \\ G_{j-1} \\ 0_{s_j,s} \end{pmatrix}.$$
    Call the 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 $\|G_j\|_F < \varepsilon$ **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

cell, C is given by

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

$e_p = [0, \ldots, 0, 1]^T$, and W^T is an n_{cell} by $(n_{eqn} \cdot n_{cell})$ block diagonal matrix. A choice for W^T is $W^T = C^T.B^{-1}$ where B 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 [25] propose the use of a multi-grid solver which is efficient in terms of iterations number but lacks scalability. 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 following. We follow the notations of the CPR preconditioner and let B be a right hand. 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

$$M_1^{-1}V_1 = C(W^TAC)^{-1}W^TV_1.$$

This application is performed by solving the system

$$(W^TAC)X = W^TV_1$$

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

$$(W^TAC)^{-1}W^TV_1$$

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

$$(W^TAC)X = W^TV_1.$$

Solving this first level by using RD-EGMRES is natural since as mentioned the right-hand side W^TV_1 has the enlarged form. Once we obtain the approximate solution X, we extend it to the second level by multiplying it with C. We save in memory the deflation preconditioner M_{def} to use it in next iterations. Then, we continue the rest of the iteration on the second level. To reapply the first level preconditioner in next iterations, we benefit from the deflation of eigenvalues that we obtained in the first iteration M_{def}. Numerical results in Section 6 show that a very fast convergence is achieved on both levels.

We note that it is possible to use EGMRES method only on the pressure level. Whereas, GMRES (or FGMRES) is used on the saturation level. In application, we consider EGMRES as a flexible preconditioner on the pressure level. Thus, the stopping criterion can be a fixed number of iterations or a small 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 breakdown 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

Considered matrices Table 1 arise from the discretization of four types of challenging problems: simulation of reservoirs, seismic imaging, linear elasticity and diffusion problems [1, 9, 20]. All numerical experiments are done by using Matlab 2016R. If it is not precisely mentioned, results correspond to RD-EGMRES.

The matrices SKY3D and ANI3D arise from boundary value problem of the diffusion equation:

$$-\text{div}(\kappa(x)\nabla u) = f \qquad \text{in } \Omega, \tag{25}$$

$$u = 0 \qquad \text{on } \partial\Omega_D, \tag{26}$$

$$\frac{\partial u}{\partial n} = 0 \qquad \text{on } \partial\Omega_N, \tag{27}$$

where Ω is the unit cube (3D). The tensor κ is a given coefficient of the partial differential operator. $\partial\Omega_D = [0,1] \times \{0,1\} \times [0,1]$. $\partial\Omega_N$ is chosen as $\partial\Omega_N = \partial\Omega \setminus \partial\Omega_D$. 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 order 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 3D, discretized on a cartesian grid of size $20 \times 20 \times 20$. The Elasticity3D100 matrix arise from the linear elasticity problem with Dirichlet and Neumann boundary conditions defined as follows

$$\text{div}(\sigma(u)) + f = 0 \qquad \text{in } \Omega, \tag{28}$$

$$u = 0 \qquad \text{on } \partial\Omega_D, \tag{29}$$

$$\sigma(u) \cdot n = 0 \qquad \text{on } \partial\Omega_N, \tag{30}$$

Ω is a unit square (2D) or a unit cube (3D). The matrices Elasticity3D100 and Elasticity2D150 correspond to this equation discretized using a triangular mesh with $100 \times 10 \times 10$ vertices for the (3D) case and 150×10 vertices for the (2D) case. $\partial\Omega_D$ is the Dirichlet boundary, $\partial\Omega_N$ is the Neumann boundary, f is a force, u is the unknown displacement field. $\sigma(.)$ is the Cauchy stress tensor given by Hooke's law: it can be expressed in terms of Young's Modulus E and Poisson's ration ν. n denotes the exterior normal vector to the boundary of Ω. For a more detailed description of the problem see [18] and [10]. We consider discontinuous E and ν: $(E_1, \nu_1) = (2 \times 10^{11}, 0.25)$ and discontinuous E in (2D): $(E_1, \nu_1) = (10^{12}, 0.45)$ and $(E_2, \nu_2) = (2 \times 10^6, 0.45)$. For the matrices BIGCO24 and BIGP1 they were obtained from the Total in-house prototype simulator for complex EOR mechanism. This simulator relies on a finite volume discretization and a two points 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 cells. BIGCO24 corresponds to a simulation of water and gaz 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. Matrices arising from reservoirs simulations and linear elasticity have one right-hand side. Seismic imaging system have multiple right-hand sides.

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

Table 1: Matrices used for tests. N is the size of the matrix, NnZ is the number of nonzero elements. HPD stands for Hermitian Positive Definite. κ is the condition number related to the second norm.

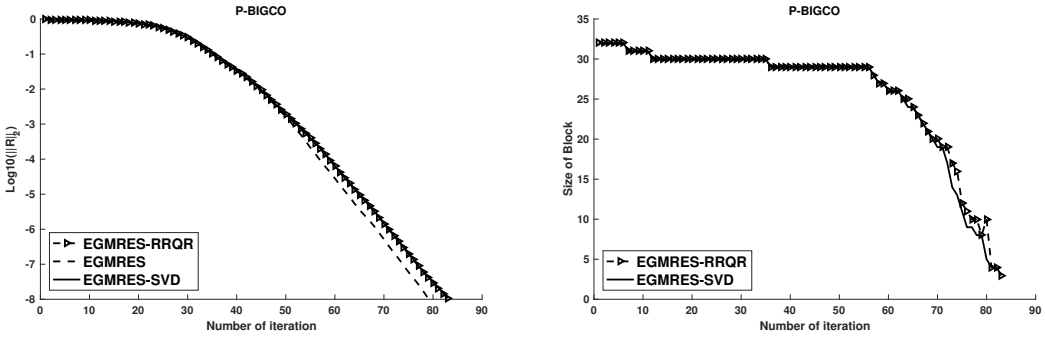


Figure 1: On the left, the convergence of EGMRES with RRQR and SVD strategies to reduce the size of block vector. On the right, impact of inexact breakdown detection on the block vector size by using each strategy.

A 128-block Jacobi preconditioner is used in all our experiment tests that we use. 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 factor of enlarging 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 number of iteration by a factor of 12 with the matrix Elasticity2D. Figures 1 and 2 show that using $RRQR$ or SVD tests to detect the inexact breakdown does not affect the robustness of the 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 up to 61% with pressure system P-BIG0 ($EF = 32$) and up to 77% with Elasticity3D100 ($EF = 16$) is obtained. We notice also in Figure 1 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 by adding 3 vectors while maintaining the rate of convergence as if 32 vectors were added at each iteration.

Matrix	It_G	EF	It_{EG}	It_{EG-SVD}	$It_{EG-RRQR}$
P-BIGCO	269	4	176	179	179
		8	141	143	143
		16	106	110	109
		32	79	83	83
P-BIGP	729	4	382	388	385
		8	273	276	276
		16	193	199	199
		32	126	129	129
Elasticity3d100	598	4	213	215	215
		8	152	157	155
		16	109	112	113
		32	80	83	83
Elasticity2d125	1490	4	466	468	470
		8	277	293	290
		16	172	180	176
		32	117	123	122
ANI3d	84	4	77	77	78
		8	72	73	73
		16	67	70	69
		32	62	64	62
SKY3d	309	4	189	191	191
		8	124	129	129
		16	70	71	71
		32	45	45	45

Table 2: Comparison between inexact breakdown detection methods. It_{Method} stands for the number of iterations to achieve convergence. *G*: GMRES as standard method. *EG*: EGMRES without inexact breakdown detection. *EG-SVD*: EGMRES with inexact breakdown detection using the SVD, as presented in [3], and *EG-RRQR*: EGMRES with inexact breakdown detection using rank revealing test (see Section 4.2). EF is the enlarging factor. Preconditioner: 128 block Jacobi. Threshold of convergence is 10^{-8}.

Matrix	EF	It_{EG250}		It_{EG500}		It_{EG750}	
		μ_1	μ_2	μ_1	μ_2	μ_1	μ_2
P-BIGCO	1	270	271	269	269	269	269
	4	187	188	181	182	179	179
	8	155	163	148	154	145	150
	16	137	160	116	132	112	125
	32	365	408	97	137	89	118
P-BIGP	1	738	740	732	733	729	729
	4	442	424	395	398	392	397
	8	+	344	299	310	288	292
	16	+	603	256	254	224	228
	32	+	+	435	255	178	186
SKY3D	1	324	324	309	309	309	309
	4	238	240	211	212	214	200
	8	197	207	160	161	146	146
	16	145	194	98	101	82	83
	32	+	+	70	73	62	68
ANI3D	1	84	84	84	84	84	84
	4	79	79	78	78	78	78
	8	75	78	74	76	73	73
	16	75	76	72	78	71	72
	32	98	95	69	73	67	76
Elasticity2D	1	1580	1551	1524	1525	1514	1516
	4	723	552	491	503	484	492
	8	+	424	330	367	317	311
	16	+	1056	241	333	228	223
	32	+	+	316	408	222	351
Elasticity3D	1	606	658	600	631	598	598
	4	226	293	218	248	216	218
	8	180	475	169	215	160	189
	16	174	363	139	176	116	157
	32	336	552	119	299	161	239

Table 3: Comparison between two tolerance values of residual eigenvectors, $\mu_1 = 5 \times 10^{-2}$, $\mu_2 = 10^{-2}$. It_{Method} stands for 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}.

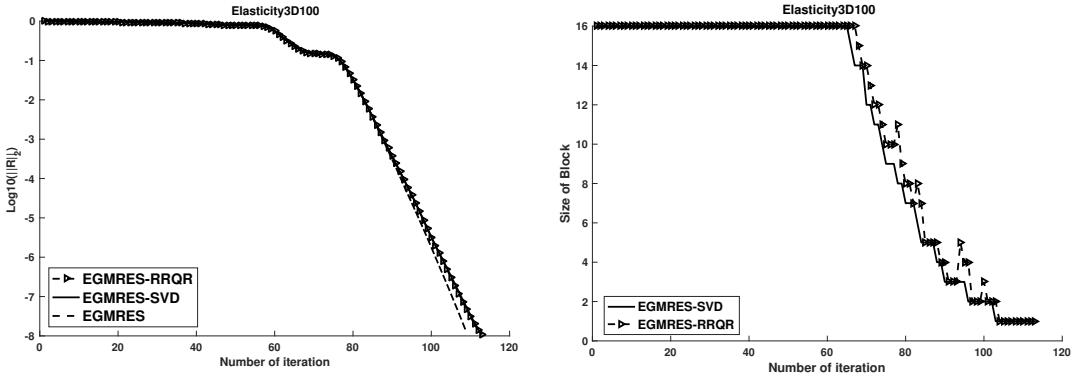


Figure 2: On the left, the convergence of EGMRES with RRQR and SVD strategies to reduce the size of block vector. On the right, impact of inexact breakdown detection on the block vector size by using each strategy.

Table 3 outlines the effect of eigenvalues deflation and the accuracy of eigenvectors estimation 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, 750}. 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 Algorithm 5) of order 10^{-2} leads to a fast convergence and maintains the speed of the convergence as if no restart is done. For example, the challenging matrix P-BIGP with $EF = 8$. EGMRES without restart needs 273 iterations to achieve convergence, while RD-EGMRES(500) needs 299 iterations. Comparing to GMRES that iterates 729 times with no restart, this difference is small. For our tests, using a threshold $\mu = 5 \times 10^{-2}$ is efficient in most cases. Choosing a larger threshold leads to a larger number of eigenvectors being deflated. This yields to less iterations per cycle that are not enough to reach convergence fast. A smaller threshold results into a small number of eigenvectors being deflated. Thus, we observe a stagnation of the residual. This results into a 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 most cases, comparing to the full method where no restart is done, RD-EGMRES keeps the rate of convergence. For some cases, when the factor of enlarging is big and the maximal size of basis is small, the method fails to converge. This occurs since the number of iterations per cycle is very few. For example, the matrix Elasticity2D with enlarging factor 32 and a maximal size of basis vectors 250 does not converge. Indeed, RD-EGMRES(250) does less than 7 iterations per cycle. That was not enough to maintain the efficiency. However, with the same size of basis and an enlarging factor $EF = 4$ a gain of factor 3 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 Figure 3, 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 factor of enlarging decreases the number of iterations. This robustness is due to the richness of the enlarged Krylov subspace and the deflation of eigenvalues (see Figures 4 and 5). The number of iterations is reduced by a factor of 3 with an enlarging factor of 16. Furthermore, we also display the impact of inexact breakdown detection on the size of the block

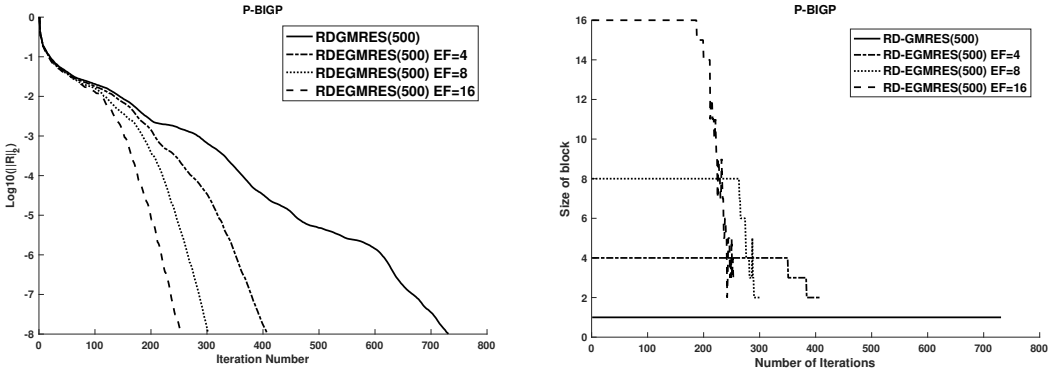


Figure 3: On the left, RD-EGMRES convergence with different enlarging factors. On the right, impact of inexact breakdown detection, based on RRQR criterion, on the size of the block vectors.

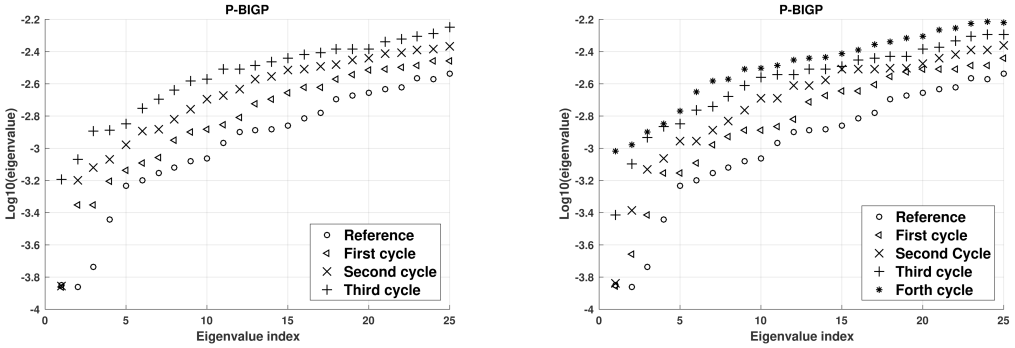


Figure 4: Eigenvalues deflation for P-BIGP. On the left $EF = 1$, on the right $EF = 4$. Preconditioner: 128 block Jacobi.

vectors. Up to an enlarging factor 16, RD-EGMRES requires approximately the same number of iterations as the full method (EGMRES) needs to reach convergence. However, the cost of orthogonalization is reduced drastically.

Figures 4 and 5 show the efficiency of RD-EGMRES to deflate eigenvalues along cycles. We compute the 25^{th} 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 25^{th} smallest eigenvalues of the deflated system. We do the same for deflated and restarted GMRES with the same size of the basis 250. We compare with the reference. RD-EGMRES(250) deflates better than the restarted GMRES. On the left of (Figure 4), deflated and restarted GMRES reaches convergence after three cycles without deflating the smallest eigenvalue, whereas on the right, RD-EGMRES(250) with $EF = 4$ deflates the smallest eigenvalue in 4 cycles. The first 4 cycles of RD-EGMRES(250) with $EF = 4$ perform sparse-matrix applications at most the same number of these type of operations that RD-GMRES(250) performs in the first cycle. In Figure 5 the test matrix, elasticity3D100, preconditioned by 128 block Jacobi, has a condition number $\kappa = 2e6$. RD-GMRES(250) converges after 2 cycles without deflating the smallest eigenvalue. RD-EGMRES(250) with $EF = 4$ and $EF = 8$ deflate all eigenvalues less than the threshold chosen, 510^{-2}, after 3 and 4 cycles

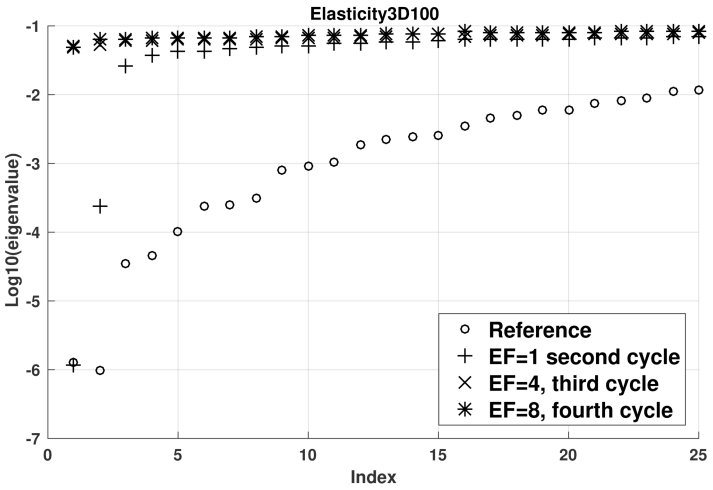


Figure 5: Comparaison: eigenvalues deflation over cycles with different enlarging factors. Eigenvalues deflation on Elasticity3D100. Preconditioner: 128 block Jacobi.

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 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.

respectively.

Table 4 shows results for seismic imaging problems with N_p right-hand sides. Enlarging the block Krylov subspace results into a 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 $9e3$, $6e4$ and $1e5$ 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 CPR preconditioner. Reservoir simulations have a structure of coupled systems. The pressure system, appearing as a submatrix, and the global system standing for the saturation system. The CPR-Preconditioning technique [25] is widely used for such problems. The main operation while solving the saturation system, by using 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

Matrix	EF	$It_{EG}B_1$	N_{ev}	$It_{EG}B_2$
P-BIGCO24	1	269	0	269
	4	181	68	117
	8	148	123	77
	16	116	156	63
P-BIGP	1	732	71	496
	4	395	182	192
	8	299	276	133
	16	256	381	155

Table 5: Influence of the factor of enlarging the Krylov subspace with multiple right-hand sides, each given 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.

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 what 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 to a linear system of equations with multiple right-hand sides each given at the application of the preconditioner. Figure 6 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 as for the factor $EF = 8$. Smallest eigenvalues have been deflated for both cases, such that improving further the conditioning is not useful. With $EF = 16$ the number of deflated vectors is 381 while it is 276 with $EF = 8$. In the end, this leads to less than eight iterations per cycle when $EF = 16$. However, no stagnation of the residual occurs, in contrast to RD-GMRES. That is related to the comparison between RD-GMRES and RD-EGMRES about the deflation of smallest eigenvalues.

Two practical approaches for using the CPR-EGMRES preconditioner, precisely on the pressure level, are proposed here:

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

In the first type, we do not need to use a flexible form while in 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 CPR preconditioner by using a direct LU solver in the level of pressure. This is the theoretical CPR approach (Section 5). In our experiment tests we compare two fixed number of iterations with the second approach 5 and 10. RD-EGMRES with the CPR preconditioner in the two levels of the system converges faster than the previously described CPR-GMRES solver. Both of previously mentioned approaches results into less number of iterations to reach convergence than the theoretical CPR-GMRES. A gain up to 50% for the non-flexible variant with $EF = 16$, matrix BIGCO24, and up to 35% for the flexible variant with $EF = 16$, matrix BIGP1. We have to mention that using the standard GMRES, either in block or simple

[1]The first iteration on the second level, RD-EGMRES has to do sufficient number of iterations, on the first level, to get deflation information.

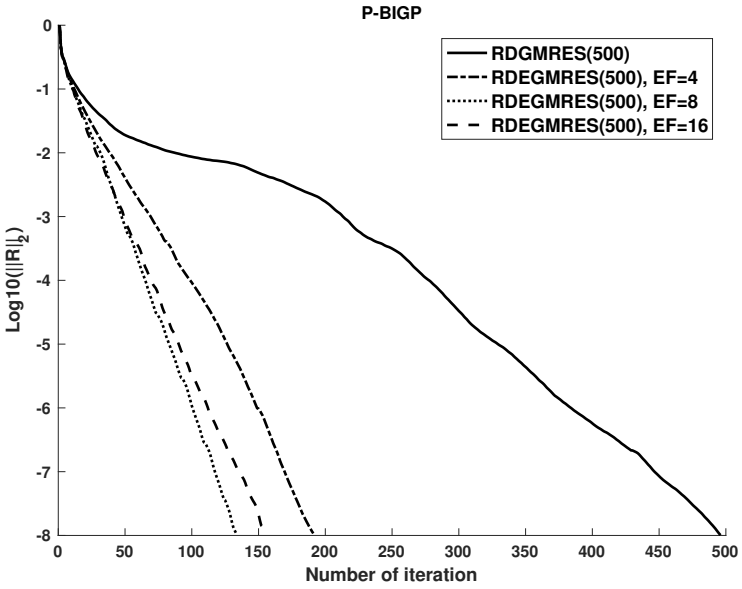


Figure 6: Deflation of linear system of equations with multiple right-hand sides each given at a time. Results show the impact of the enlarging factor on the convergence of $AX = B_2$ and a comparison with RD-GMRES(500). $AX = B_1$ was previously solved by using RD-GMRES(500) and a deflation preconditioner is constructed to solve $AX = B_2$.

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}		

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}.

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. That explains why the error in the solution in Table 6 for non-flexible methods is far from the error related to the flexible variant.

7 Conclusion

Based on two previous works, robust general linear solver, GMRES [24], and a communication reducing approach, enlarged Krylov subspace [9], we introduced Enlarged GMRES, a communication reducing robust general linear solver. At each iteration of Enlarged GMRES, 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 into a faster convergence. Due to limited memory and the cost of orthogonalization flops, restarting the method is necessary. One solution to this problem is the reduction of the added block vectors, once they are not useful. It maintains the rate of convergence of the method, as if no reduction is done, and decreases the computational and memory cost. Starting from the theory of exact and inexact breakdown introduced in [22], we developed a new theoretical and practical strategy to detect inexact breakdowns based on rank revealing QR of a squared matrix of size T 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 for all block GMRES-like methods.

The necessity to solve linear systems of equations with multiple right-hand sides, each given 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, originally presented in [7]. This theorem gives an algebraic formulation for a preconditioner once we have the approximate eigenvectors. Thus, we proposed an approach, based on 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 of 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 simulation of reservoir. Unlike existing methods, such as proposed in [25], where an algebraic multi grid solves the second level and FGMRES solves the first level, 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 of up to 2 is obtained by CPR-EGMRES compared to CPR-GMRES with ideal conditions, i.e. when a direct linear solver is used in the second level. In conclusion, EGMRES reduces the number of iterations to reach convergence. This method is addressed for ill-conditioned linear systems. We compared different thresholds for the criteria of eigenvectors approximation. We noticed that a threshold $\varepsilon = 5 \times 10^{-2}$ leads to good results in general. As future work, the method will be implemented on parallel machine and larger test cases will be tested on massively parallel computers.

References

[1] Y. Achdou and F. Nataf, *Low frequency tangential filtering decomposition*, Numerical Linear Algebra with Applications, 14 (2007), pp. 129–147.

[2] E. Agullo, L. Giraud, and Y.-F. Jing, *Block gmres method with inexact breakdowns and deflated restarting*, SIAM Journal on Matrix Analysis and Applications, 35 (2014), pp. 1625–1651.

[3] H. Calandra, S. Gratton, R. Lago, X. Vasseur, and L. M. Carvalho, *A modified block flexible gmres method with deflation at each iteration for the solution of non-hermitian linear systems with multiple right-hand sides*, SIAM Journal on Scientific Computing, 35 (2013), pp. S345–S367.

[4] T. F. Chan, *Rank revealing qr factorizations*, Linear Algebra and its Applications, 88 (1987), pp. 67 – 82.

[5] A. Chronopoulos and C. Gear, *s-step iterative methods for symmetric linear systems*, Journal of Computational and Applied Mathematics, 25 (1989), pp. 153 – 168.

[6] A. el Guennouni, K. Jbilou, and H. Sadok, *A block version of bicgstab for linear systems with multiple right-hand sides.*, ETNA. Electronic Transactions on Numerical Analysis [electronic only], 16 (2003), pp. 129–142.

[7] J. Erhel, K. Burrage, and B. Pohl, *Restarted gmres preconditioned by deflation*, Journal of Computational and Applied Mathematics, 69 (1996), pp. 303 – 318.

[8] R. W. Freund and M. Malhotra, *A block qmr algorithm for non-hermitian linear systems with multiple right-hand sides*, Linear Algebra and its Applications, 254 (1997), pp. 119 – 157.

[9] L. Grigori, S. Moufawad, and F. Nataf, *Enlarged Krylov Subspace Conjugate Gradient Methods for Reducing Communication*, Research Report RR-8597, INRIA, Sept. 2014.

[10] L. Grigori, F. Nataf, and S. Yousef, *Robust algebraic Schur complement preconditioners based on low rank corrections*, Research Report RR-8557, INRIA, July 2014.

[11] M. Gutknecht and T. Schmelzer, *Updating the qr decomposition of block tridiagonal and block hessenberg matrices generated by block krylov space methods*, Appl. Num. Math, 85 (2008), pp. 871–883.

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

[13] M. R. Hestenes and E. Stiefel., *Methods of conjugate gradients for solving linear systems.*, Journal of research of the National Bureau of Standards., 49 (1952), pp. 409–436.

[14] M. Hoemmen, *Communication-Avoiding Krylov Subspace Mehtods.*, PhD thesis, EECS Department, University of California, Berkeley, 2010.

[15] J. Langou, *Iterative methods for solving linear systems with multiple right-hand sides*, PhD thesis, CERFACS, 2003.

[16] R. B. Morgan, *A restarted gmres method augmented with eigenvectors*, SIAM Journal on Matrix Analysis and Applications, 16 (1995), pp. 1154–1171.

[17] R. B. Morgan, *Restarted block-gmres with deflation of eigenvalues*, Appl. Numer. Math., 54 (2005), pp. 222–236.

[18] F. Nataf, F. Hecht, P. Jolivet, and C. Prud'Homme, *Scalable domain decomposition preconditioners for heterogeneous elliptic problems*, SC13, (Denver, Colorado, United States), (2013).

[19] A. A. Nikishin and A. Y. Yeremin, *Variable block cg algorithms for solving large sparse symmetric positive definite linear systems on parallel computers, i: General iterative scheme*, SIAM Journal on Matrix Analysis and Applications, 16 (1995), pp. 1135–1153.

[20] Q. Niu, L. Grigori, P. Kumar, and F. Nataf, *Modified tangential frequency filtering decomposition and its fourier analysis*, Numerische Mathematik, 116 (2010), pp. 123–148.

[21] D. P. O'Leary, *The block conjugate gradient algorithm and related methods*, Linear Algebra and its Applications, 29 (1980), pp. 293 – 322.

[22] M. Robbé and M. Sadkane, *Exact and inexact breakdowns in the block gmres method*, Linear Algebra and its Applications, 419 (2006), pp. 265 – 285.

[23] Y. Saad, *Iterative Methods for Sparse Linear Systems*, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2nd ed., 2003.

[24] Y. Saad and M. H. Schultz, *Gmres: A generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM Journal on Scientific and Statistical Computing, 7 (1986), pp. 856–869.

[25] H. C. S. Schlumberger, H. T. S. S. U., J. W. W. C. Inc., and H. Y. S. Chevron, *Parallel scalable unstructured CPR-Type Linear Solver for Resevoir Simulation*, SPE 96809, (2005).

[26] V. Simoncini and E. Gallopoulos, *Convergence properties of block gmres and matrix polynomials*, Linear Algebra and its Applications, 247 (1996), pp. 97 – 119.

[27] P. Sonneveld, *Cgs, a fast lanczos-type solver for nonsymmetric linear systems*, SIAM Journal on Scientific and Statistical Computing, 10 (1989), pp. 36–52.

[28] J. M. Tang, R. Nabben, C. Vuik, and Y. A. Erlangga, *Comparison of two-level preconditioners derived from deflation, domain decomposition and multigrid methods*, Journal of Scientific Computing, 39 (2009), pp. 340–370.

[29] H. A. van der Vorst, *Bi-cgstab: A fast and smoothly converging variant of bi-cg for the solution of nonsymmetric linear systems*, SIAM Journal on Scientific and Statistical Computing, 13 (1992), pp. 631–644.

[30] B. Vital, *Étude de quelques méthodes de résolution de problèmes lineaires de grande taille sur multiprocesseur*, PhD thesis, Rennes 1, 1990.

[31] J. Wallis, R. Kendall, and T. Little, *Constrained residual acceleration of conjugate residual methods*, SPE 13563, (1985).

A Update of QR factorization of the Hessenberg matrix

In this appendix we explain under the form of a constructive proof of Lemma 2 how we update the QR factorization of the matrix $\tilde{H}$. In [11] 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.

Lemma 3. *Lemma 2*

Proof. 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 (8),

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

Relation (10) and the induction hypothesis give the QR factorization of H_j

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

We can write

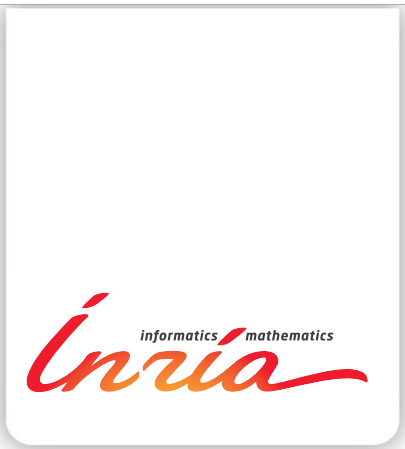
$$\tilde{H}_{j+1} = \left(\prod_{i=0}^{j-1} \mathcal{Q}^H_{(j+1-i),(j+1)}\right) \left(\prod_{i=1}^{j} \mathcal{F}_{i,(j+1)}\right) \begin{pmatrix} \begin{pmatrix} C_j \\ 0_{s,S_j} \end{pmatrix} & \begin{pmatrix} n_{j+1,1} \\ n_{j+1,2} \end{pmatrix} \\ 0_{s_j,S_j} & M_{j+1} \end{pmatrix}.$$

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

Let F_{j+1} be the matrix of Housholder transormation that triangularize $\begin{pmatrix} n_{j+1,2} \\ M_{j+1} \end{pmatrix}$, then we obtain the relation satisfied for $j + 1$

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

□



RESEARCH CENTRE
PARIS – ROCQUENCOURT

Domaine de Voluceau, - Rocquencourt
B.P. 105 - 78153 Le Chesnay Cedex

Publisher
Inria
Domaine de Voluceau - Rocquencourt
BP 105 - 78153 Le Chesnay Cedex
inria.fr

ISSN 0249-6399