Available online at www.sciencedirect.com

ScienceDirect

Linear Algebra and its Applications 419 (2006) 265–285

LINEAR ALGEBRA AND ITS APPLICATIONS

www.elsevier.com/locate/laa

Exact and inexact breakdowns in the block GMRES method

Mickaël Robbé *, Miloud Sadkane

Université de Bretagne Occidentale, Département de Mathématiques, 6 Av. Le Gorgeu, CS 93837, 29238 Brest Cedex 3, France

Received 21 October 2004; accepted 21 April 2006
Available online 14 July 2006
Submitted by V. Mehrmann

Abstract

This paper addresses the issue of breakdowns in the block GMRES method for solving linear systems with multiple right-hand sides of the form $AX = B$. An exact (inexact) breakdown occurs at iteration j of this method when the block Krylov matrix $(B, AB, \ldots, A^{j-1}B)$ is singular (almost singular). Exact breakdowns are the sign that a part of the exact solution is in the range of the Krylov matrix. They are primarily of theoretical interest. From a computational point of view, inexact breakdowns are most likely to occur. In such cases, the underlying block Arnoldi process that is used to build the block Krylov space should not be continued as usual. A natural way to continue the process is the use of deflation. However, as shown by Langou [J. Langou, Iterative methods for solving linear systems with multiple right-hand sides, Ph.D. dissertation TH/PA/03/24, CERFACS, France, 2003], deflation in block GMRES may lead to a loss of information that slows down the convergence. In this paper, instead of deflating the directions associated with almost converged solutions, these are kept and reintroduced in next iterations if necessary. Two criteria to detect inexact breakdowns are presented. One is based on the numerical rank of the generated block Krylov basis, the second on the numerical rank of the residual associated to approximate solutions. These criteria are analyzed and compared. Implementation details are discussed. Numerical results are reported.

AMS classification: 65F10

Keywords: Block GMRES; Breakdown

* Corresponding author.
E-mail addresses: robbe@univ-brest.fr (M. Robbé), sadkane@univ-brest.fr (M. Sadkane).

0024-3795/$ - see front matter
doi:10.1016/j.laa.2006.04.018

1. Introduction

The purpose of this paper is to analyze the issue of breakdown in the block GMRES method (referred to as BGMRES hereafter) for the solution of linear systems with multiple right-hand sides

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

where $A \in \mathbf{C}^{N\times N}$ is a non-singular matrix of large dimension N and $B \in \mathbf{C}^{N\times p}$ is full rank with $p \ll N$. Variants of BGMRES are discussed for example in [14,13,12,6,10,8]. See also the recent bibliography in [5]. Starting with a zero initial guess, the method finds approximations to $X = A^{-1}B$ of the form

$$X_j = \mathscr{V}_j Y_j, \tag{2}$$

where $\mathscr{V}_j$ is a matrix whose columns form an orthonormal basis of the block Krylov space

$$\mathscr{K}_j(A, B) = \text{Range}\begin{pmatrix} B & AB & \cdots & A^{j-1}B \end{pmatrix}. \tag{3}$$

The construction of $\mathscr{V}_j$ is accomplished with the help of a block version of Arnoldi's algorithm [1,12] which simultaneously constructs a rectangular block-Hessenberg matrix $\underline{\mathscr{H}}_j$ such that the equality $A\mathscr{V}_j = \mathscr{V}_{j+1}\underline{\mathscr{H}}_j$ holds.

An exact breakdown occurs when this iteration cannot be continued. This happens when the matrix $\begin{pmatrix} B & AB & \cdots & A^{j-1}B \end{pmatrix}$ is rank-deficient. As will be seen, this situation means that the spaces Range$X$ and $\mathscr{K}_j(A, B)$ intersect at some space whose dimension equals that of the null space of the block residual $R_j = B - AX_j$. In other words, BGMRES has computed a linear combination of some columns of X.

An inexact breakdown occurs when the matrix $\begin{pmatrix} B & AB & \cdots & A^{j-1}B \end{pmatrix}$ is almost rank-deficient. Unlike the case of exact breakdown, this situation is most likely to happen in practice. In such a situation, the block Arnoldi algorithm (see Algorithm 1) that is used to construct the block Krylov basis should not be continued as such, and some modifications are required.

The issue of breakdowns in block Krylov methods is not new. In [9], Nikishin and Yeremin propose a criterion to detect inexact breakdowns based on a block version of the CG method. The authors show that the directions associated with the sufficiently small singular values of the block residual have almost converged. They delete these vectors from subsequent iterations and continue the process until all vectors of the block residual are deleted. The breakdowns are also encountered in the QMR method [2] which uses the non-symmetric Lanczos algorithm [7]. Here, exact or near breakdowns may have a different meaning. They may be caused by a division by zero or near zero during the Lanczos iterations. These breakdowns are remedied by incorporating look-ahead into the algorithm. In [3], Freund and Malhotra exclude such breakdowns. Their implementation allows to detect the linearly or almost linearly dependent vectors in the generated block Krylov spaces. Detection of such vectors signals that a linear combination of some columns of X has converged. Then, one or several columns of B are deleted and the process continues. The process of deletion is referred to as deflation in [3]. In the context of BGMRES, Langou [8, p. 210] has shown that deflation is not recommended since it may lead to a loss of information that slows down the convergence.

In this paper, we do not delete (deflate) the inexact breakdowns associated with almost converged solutions obtained by BGMRES. We instead keep these vectors and reintroduce them in next iterations if necessary.

The advantage of keeping these vectors, at a given iteration, is the possibility of using a weak condition to detect them since, as we will see, even if they are no longer considered as inexact breakdowns in a next iteration, they can be reintroduced in the block Krylov space to obtain a good enough solution. The quick detection of inexact breakdowns reduces the computational cost and the storage requirements whereas the reintroduction of inexact breakdown vectors improves the quality of the computed solution. Of course, inexact breakdowns are defined with the help of a threshold parameter. The main difficulty is to select, at the right iteration with the right criterion, the inexact breakdowns which allow to compute at the final iteration, a good enough solution. If inexact breakdowns are selected under a weak condition then BGMRES generally provides a fast but less accurate solution. On the other hand, a strong condition may lead to useless computations.

We analyze and compare two strategies to detect inexact breakdowns. One strategy is based on the numerical rank of the generated basis, the second on the numerical rank of the residuals associated to approximate solutions. As we will see the latter strategy is more efficient than the former.

The plan of the paper is as follows. In Section 2 we characterize the exact breakdowns in the block Arnoldi algorithm and discuss their effects on BGMRES. In Section 3, the perturbations introduced by inexact breakdowns are analyzed for the block Arnoldi algorithm and BGMRES. Here and throughout this paper, we assume that exact arithmetic is used. In particular we assume that inexact breakdowns are not the result of a finite precision arithmetic. Section 4 is devoted to the detection of inexact breakdowns and the consequences on the quality of the computed solution. Implementation details of BGMRES with the treatment of inexact breakdowns are described in Section 5. Section 6 illustrates and compares the proposed strategies on some numerical examples.

The symbol $\|\cdot\|_q$ denotes the Euclidean matrix norm when $q = 2$ and the Frobenius norm when $q = F$. The notation x^* is used for complex (and real) cases to denote the transpose conjugate of x. The identity and null matrices of order k are denoted respectively by I_k or 0_k or just I and 0 when the order is evident from the context. If $C \in \mathbf{C}^{k\times l}$, we denote the singular values of C by $\sigma_1(C) \geqslant \cdots \geqslant \sigma_{\min(k,l)}(C)$.

2. Block Arnoldi and exact breakdowns

Let $A \in \mathbf{C}^{N\times N}$ and $V_1 \in \mathbf{C}^{N\times p_1}$ with $p_1 \ll N$ and $V_1^* V_1 = I_{p_1}$. An orthonormal basis of the block Krylov space

$$\mathscr{K}_j(A, V_1) = \mathrm{Range}\begin{pmatrix} V_1 & AV_1 & \cdots & A^{j-1}V_1 \end{pmatrix} \tag{4}$$

is constructed using the following algorithm

Algorithm 1 (*Block Arnoldi*)

For $j = 1, \ldots, m$

- $W_j = AV_j$

For $i = 1, \ldots, j$

- $H_{i,j} = V_i^* W_j$
- $W_j = W_j - V_i H_{i,j}$

End For i

- $W_j = V_{j+1} H_{j+1,j}$ (reduced QR factorization)

End For j

From now on, W_j denotes the matrix at the end of the i-loop, that is, the matrix of which the reduced QR factorization is computed. We denote by p_{j+1} the rank of V_{j+1}, thus $V_{j+1} \in \mathbf{C}^{N\times p_{j+1}}$,

$W_j \in \mathbf{C}^{N \times p_j}$ and $H_{j+1,j} \in \mathbf{C}^{p_{j+1} \times p_j}$. Since the reduced QR factorization is applied to W_j, we have $p_{j+1} = \text{rank}(W_j) = \text{rank}(H_{j+1,j})$.

Using the notation

$$n_j = \sum_{i=1}^{j} p_i, \tag{5}$$

$$\mathscr{V}_j = \begin{pmatrix} V_1 & \cdots & V_j \end{pmatrix} \in \mathbf{C}^{N \times n_j}, \tag{6}$$

$$\mathscr{V}_{j+1} = \begin{pmatrix} \mathscr{V}_j & V_{j+1} \end{pmatrix} \in \mathbf{C}^{N \times n_{j+1}}, \tag{7}$$

$$\mathscr{H}_j = (H_{i,l})_{1 \leqslant i,l \leqslant j} \in \mathbf{C}^{n_j \times n_j}, \tag{8}$$

$$\underline{\mathscr{H}}_j = \begin{pmatrix} & \mathscr{H}_j & \\ 0 \quad \cdots & 0 & H_{j+1,j} \end{pmatrix} \in \mathbf{C}^{n_{j+1} \times n_j}, \tag{9}$$

we obtain from Algorithm 1

$$A\mathscr{V}_j = \mathscr{V}_j \mathscr{H}_j + \begin{pmatrix} 0 & W_j \end{pmatrix} = \mathscr{V}_{j+1} \underline{\mathscr{H}}_j. \tag{10}$$

Moreover, the matrix $\mathscr{H}_j$ is block upper Hessenberg and the columns of $\mathscr{V}_j$ form an orthonormal basis of $\mathscr{K}_j(A, V_1)$.

It is clear that the sequence $(p_j)_{j \geqslant 1}$ is non-increasing. If $p_{j+1} = p_j = \cdots = p_1$, then no breakdown occurs. If $W_{j+1} = 0$, then $p_{j+1} = 0$ and both V_{j+1} and $H_{j+1,j}$ are not defined. We will say that Algorithm 1 has $k_j = p_1 - p_{j+1}$ exact breakdowns up to iteration j if

$$p_{j+1} < p_1. \tag{11}$$

The following theorem characterizes the property (11):

Theorem 1. *The condition* (11) *is equivalent to*

$$\dim\{\text{Range} V_1 \cap A\mathscr{K}_j(A, V_1)\} = p_1 - p_{j+1} > 0, \tag{12}$$

where $A\mathscr{K}_j(A, V_1) = \text{Range}\begin{pmatrix} AV_1 & A^2V_1 & \cdots & A^jV_1 \end{pmatrix}$.

Proof

$$\begin{aligned} &\dim\{\text{Range} V_1 \cap A\mathscr{K}_j(A, V_1)\} \\ &\quad = \dim\{\text{Range} V_1\} + \dim\{A\mathscr{K}_j(A, V_1)\} - \dim\{\text{Range} V_1 + A\mathscr{K}_j(A, V_1)\} \\ &\quad = \dim\{\text{Range} V_1\} + \dim\{\mathscr{K}_j(A, V_1)\} - \dim\{\mathscr{K}_{j+1}(A, V_1)\} \\ &\quad = p_1 + n_j - n_{j+1} = p_1 - p_{j+1}. \qquad \square \end{aligned}$$

2.1. Effects of exact breakdowns on BGMRES

Suppose we want to solve the block system (1). Let $B = V_1 \Lambda_1$ the QR factorization of B, where $V_1 \in \mathbf{C}^{N \times p}$ and $\Lambda_1 \in \mathbf{C}^{p \times p}$ are full rank. Starting with V_1 and denoting $p_1 = p$, Algorithm 1 constructs matrices $H_{i,l} \in \mathbf{C}^{p_i \times p_l}$ and V_l with the properties (5)–(10).

BGMRES yields an approximation to $X = A^{-1}B$ of the form

$$X_j = \mathscr{V}_j Y_j, \tag{13}$$

where Y_j solves the block linear system $(A\mathscr{V}_j)^*(V_1\Lambda_1 - A\mathscr{V}_jY_j) = 0$, which can be written

$$\min_{Y\in\mathbf{C}^{n_j\times p_1}} \left\| \underline{\mathscr{H}}_j Y - \underline{\Lambda}_j \right\|_F \quad \text{with } \underline{\Lambda}_j = \begin{pmatrix}\Lambda_1\\0\end{pmatrix} \in \mathbf{C}^{n_{j+1}\times p_1}. \tag{14}$$

Note that since $\underline{\mathscr{H}}_j$ has full columns rank, the least squares problem (14) has the unique solution $Y_j = \underline{\mathscr{H}}_j^\dagger \underline{\Lambda}_j$ where $\underline{\mathscr{H}}_j^\dagger = \left(\underline{\mathscr{H}}_j^* \underline{\mathscr{H}}_j\right)^{-1} \underline{\mathscr{H}}_j^*$.

If we denote by

$$R_j = B - AX_j \tag{15}$$

the residual associated to BGMRES, then it is easy to see that

$$R_j = \mathscr{V}_{j+1}(\underline{\Lambda}_j - \underline{\mathscr{H}}_j Y_j) \tag{16}$$

and

$$\|R_j\|_F = \|\underline{\Lambda}_j - \underline{\mathscr{H}}_j Y_j\|_F = \min_{Y\in\mathbf{C}^{n_j\times p_1}} \|B - A\mathscr{V}_j Y\|_F. \tag{17}$$

Since A is non-singular, the condition (12) is equivalent to

$$\dim\{\mathrm{Range}X \cap \mathscr{K}_j(A, V_1)\} = k_j = p_1 - p_{j+1} > 0. \tag{18}$$

Therefore, k_j partial breakdowns occur if and only if the block Krylov space $\mathscr{K}_j(A, V_1)$ contains a vector space of dimension k_j formed by a linear combination of some columns of X, the solution of (1). This property is interesting for BGMRES whose solution belongs to $\mathscr{K}_j(A, V_1)$. It was used by Nikishin and Yeremin for the block CG method [9].

The following corollary shows that when the condition (12) is satisfied, the null space $X - X_j$ is of dimension k_j. Roughly speaking, this means that a part of X of dimension k_j has been computed.

Corollary 1. *The condition* (11) *and therefore the conditions* (12) *and* (18) *are equivalent to*

$$\mathrm{rank}(R_j) = p_{j+1} < p_1. \tag{19}$$

Proof. From Theorem 1, the condition (11) is satisfied if there exist full rank matrices $U_1 \in \mathbf{C}^{p_1\times(p_1-p_{j+1})}$ and $U_2 \in \mathbf{C}^{n_j\times(p_1-p_{j+1})}$ such that $BU_1 = A\mathscr{V}_jU_2$.

Then from the definition of Y_j, we have

$$0 = (A\mathscr{V}_j)^*(B - A\mathscr{V}_jY_j)U_1 = (A\mathscr{V}_j)^*(A\mathscr{V}_j)(U_2 - Y_jU_1).$$

Since A is non-singular and $\mathscr{V}_j$ is full rank, this implies that $Y_jU_1 = U_2$ and

$$R_jU_1 = BU_1 - A\mathscr{V}_jY_jU_1 = A\mathscr{V}_j(U_2 - Y_jU_1) = 0.$$

Therefore $\mathrm{rank}(R_j) \leqslant p_{j+1}$.

Now assume that (19) is satisfied and consider a full rank matrix $Z \in \mathbf{C}^{p_1\times(p_1-p_{j+1})}$ such that

$$BZ = A\mathscr{V}_jY_jZ.$$

Since B is full rank, this implies that

$$\dim\{\mathrm{Range}(V_1) \cap \mathrm{Range}(A\mathscr{K}_j(A, V_1))\} \geqslant p_1 - p_{j+1} > 0.$$

The reasoning above shows the required equivalence. □

In conclusion, exact breakdowns are favorable situations for BGMRES. They are direct signs of a partial convergence. They can be equivalently detected from the ranks of W_j or R_j. In practice, inexact breakdowns can also be detected from criteria based on W_j or the residual but as we will see, these criteria do not lead to the same conclusion and modify the behavior of BGMRES.

3. Computations with inexact breakdowns

In the exact case we have (see (10))

$$A\mathscr{V}_j = \mathscr{V}_j\mathscr{H}_j + \begin{pmatrix} 0 & W_j \end{pmatrix} \quad \text{with } W_j = V_{j+1}H_{j+1,j},$$

and exact breakdowns occur up to iteration j when $p_{j+1} = \operatorname{rank}\begin{pmatrix} 0 & W_j \end{pmatrix} < p_1$ (see (11)) or equivalently when $\operatorname{rank}(R_j) < p_1$ (see (19)).

These conditions are too severe to be satisfied in practice. For this reason, we introduce two criteria denoted hereafter W-criterion and R-criterion inspired by (11) and (19) respectively to define the notion of inexact breakdowns. Unlike the exact case, these criteria are not equivalent and may therefore detect different inexact breakdowns.

Both criteria require the following modification of the reduced QR factorization step in Algorithm 1

$$W_j = V_{j+1}H_{j+1,j} + Q_j, \tag{20}$$

where Q_j is a perturbation matrix which takes into account the numerical rank in the QR factorization. The matrix Q_j is not the result of a finite precision arithmetic. We assume that RangeV_{j+1} and RangeQ_j are orthogonal. Moreover, since by definition RangeW_j is orthogonal to Range$\mathscr{V}_j$, we see that RangeQ_j is orthogonal to Range$\mathscr{V}_{j+1}$. It is important to remember this property which will be used in the sequel.

At first sight, we may think of deleting the perturbations $Q_j's$. But this introduces errors in the block Arnoldi process that lead to inaccurate solutions (see Fig. 3).

The next (sub)sections present some properties of the block Arnoldi process when the $Q_j's$ are used and the strategies to detect inexact breakdowns. The main idea and motivations are the following: the use of $Q_j's$ modifies the Arnoldi relations (5)–(9). Indeed, if we define $\mathscr{Q}_j$ and $\widetilde{\mathscr{Q}}_j$ by

$$\mathscr{Q}_j = \begin{pmatrix} Q_1 & \cdots & Q_j \end{pmatrix} \quad \text{and} \quad \widetilde{\mathscr{Q}}_j = (I - \mathscr{V}_{j+1}\mathscr{V}_{j+1}^*)\mathscr{Q}_j, \tag{21}$$

then, instead of (10) we will have (see (25))

$$A\mathscr{V}_j = \mathscr{V}_j\mathscr{H}_j + \begin{pmatrix} \mathscr{Q}_{j-1} & W_j \end{pmatrix}. \tag{22}$$

However, this is not a block Arnoldi-like relation because $\mathscr{H}_j \neq \mathscr{V}_j^*A\mathscr{V}_j$. The good relation is (see (26))

$$A\mathscr{V}_j = \mathscr{V}_j\mathscr{L}_j + \begin{pmatrix} \widetilde{\mathscr{Q}}_{j-1} & W_j \end{pmatrix} \quad \text{with } \mathscr{L}_j \equiv \mathscr{V}_j^*A\mathscr{V}_j = \mathscr{H}_j + \mathscr{V}_j^*\mathscr{Q}_j \tag{23}$$

even though the matrix $\mathscr{L}_j$ does not have a block-Hessenberg structure (see (27)). However, to be of a practical use, the full and $N \times n_j$ matrix $\begin{pmatrix} \widetilde{\mathscr{Q}}_{j-1} & W_j \end{pmatrix}$ in (23) must satisfy a condition like

$$\operatorname{Range} V_{j+1} \subset \operatorname{Range}\begin{pmatrix} \widetilde{\mathscr{Q}}_{j-1} & W_j \end{pmatrix} \quad \text{with } \widetilde{\mathscr{Q}}_0 = 0. \tag{24}$$

First, this inclusion does not contradict (23) since both Range$\widetilde{\mathscr{Q}}_{j-1}$ and RangeW_j are orthogonal to Range$\mathscr{V}_j$. Second, it has the important consequence that, for all j, the rank of $\widetilde{\mathscr{Q}}_j$ is always less or equal to p_1 (see (29)). In practice, this means that $\widetilde{\mathscr{Q}}_j$ can be stored in an array of at most p_1 columns. Our implementation uses this property (see (54) and Section 5).

To adapt (11) and (19) for the inexact case, we will use the singular values of $\begin{pmatrix}\widetilde{\mathscr{Q}}_{j-1} & W_j\end{pmatrix}$ (the W-criterion) and those of the residual R_j obtained in the inexact case (the R-criterion). These criteria will be clarified in Section 4.

3.1. Block Arnoldi and inexact breakdowns

The inexact breakdown case requires the decomposition (20) and inclusion (24). Under these conditions, the properties of V_{j+1} remain unchanged, namely its size is $N \times p_{j+1}$, it has full rank, and its range is orthogonal to that of $\mathscr{V}_j$. However unlike the exact case, now the rank of W_j can be different from p_{j+1}, the sequence $(p_j)_{j\geqslant 1}$ does not necessarily decrease and $H_{j+1,j}$ can be rank-deficient.

The following theorem shows that $\mathscr{V}_j$ still satisfies properties analogous to those of the block Arnoldi algorithm:

Theorem 2. *We have*

$$A\mathscr{V}_j = \mathscr{V}_j\mathscr{H}_j + \begin{pmatrix}\mathscr{Q}_{j-1} & W_j\end{pmatrix} = \mathscr{V}_{j+1}\underline{\mathscr{H}}_j + \mathscr{Q}_j, \tag{25}$$

$$A\mathscr{V}_j = \mathscr{V}_j\mathscr{L}_j + \begin{pmatrix}\widetilde{\mathscr{Q}}_{j-1} & W_j\end{pmatrix} = \mathscr{V}_{j+1}\underline{\mathscr{L}}_j + \widetilde{\mathscr{Q}}_j \tag{26}$$

with

$$\mathscr{L}_j \equiv \mathscr{V}_j^* A\mathscr{V}_j = \mathscr{H}_j + \mathscr{V}_j^*\mathscr{Q}_j = \begin{pmatrix} H_{1,1} & \cdots & \cdots & \cdots & H_{1,j} \\ H_{2,1} & \ddots & & & \vdots \\ V_3^*Q_1 & \ddots & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ V_j^*Q_1 & \cdots & V_j^*Q_{j-2} & H_{j,j-1} & H_{j,j} \end{pmatrix} \tag{27}$$

and with obvious notation

$$\underline{\mathscr{L}}_j \equiv \mathscr{V}_{j+1}^* A\mathscr{V}_j = \underline{\mathscr{H}}_j + \mathscr{V}_{j+1}^*\mathscr{Q}_j. \tag{28}$$

Moreover, the rank of $\widetilde{\mathscr{Q}}_j$, *denoted by* $\tilde{q}_j$, *satisfies*

$$\tilde{q}_j + p_{j+1} \leqslant p_1 \quad \text{for } j \geqslant 0. \tag{29}$$

Proof. Algorithm 1 where the QR factorization step is replaced by (20) gives

$$AV_j = \mathscr{V}_j \begin{pmatrix} H_{1,1} \\ \vdots \\ H_{j,j} \end{pmatrix} + V_{j+1}H_{j+1,j} + Q_j, \tag{30}$$

from which follow (25)–(28).

We prove (29) by induction. Since $\widetilde{\mathscr{Q}}_0 = 0$, (29) holds for $j = 0$. Assume it is true at iteration $j-1$, then

$$\operatorname{rank}\begin{pmatrix}\widetilde{\mathscr{Q}}_{j-1} & W_j\end{pmatrix} \leqslant \operatorname{rank}\left(\widetilde{\mathscr{Q}}_{j-1}\right) + \operatorname{rank}\left(W_j\right) \leqslant \tilde{q}_{j-1} + p_j \leqslant p_1. \tag{31}$$

Now since $\mathrm{Range}(V_{j+1}) \subset \mathrm{Range}\begin{pmatrix}\widetilde{\mathcal{Q}}_{j-1} & W_j\end{pmatrix}$, we have

$$\begin{aligned}\mathrm{Range}\begin{pmatrix}\widetilde{\mathcal{Q}}_{j-1} & W_j\end{pmatrix} &= \mathrm{Range}\begin{pmatrix}V_{j+1} & (I - V_{j+1}V_{j+1}^*)\widetilde{\mathcal{Q}}_{j-1} & (I - V_{j+1}V_{j+1}^*)W_j\end{pmatrix}\\ &= \mathrm{Range}\begin{pmatrix}V_{j+1} & \widetilde{\mathcal{Q}}_j\end{pmatrix}.\end{aligned}$$

The equalities above are justified by $(I - V_{j+1}V_{j+1}^*)W_j = Q_j$, $V_{j+1}^*Q_j = 0$ and $V_{j+1}^*\widetilde{\mathcal{Q}}_j = 0$. Thus we have

$$\mathrm{rank}\begin{pmatrix}\widetilde{\mathcal{Q}}_{j-1} & W_j\end{pmatrix} = \mathrm{rank}(V_{j+1}) + \mathrm{rank}(\widetilde{\mathcal{Q}}_j) = p_{j+1} + \tilde{q}_j. \tag{32}$$

The proof follows then from (31) and (32). □

The following proposition used in Section 4, shows that $\underline{\mathscr{L}}_j$ and $\widetilde{\mathcal{Q}}_j$ can be computed iteratively.

Proposition 1. *We have*

$$\mathscr{L}_j = \begin{pmatrix} & H_{1,j}\\ \underline{\mathscr{L}}_{j-1} & \vdots\\ & H_{j,j}\end{pmatrix}, \quad \underline{\mathscr{L}}_j = \begin{pmatrix}\mathscr{L}_j\\ V_{j+1}^*\widetilde{\mathcal{Q}}_{j-1}H_{j+1,j}\end{pmatrix}, \tag{33}$$
$$\textit{and } \widetilde{\mathcal{Q}}_j = \begin{pmatrix}(I - V_{j+1}V_{j+1}^*)\widetilde{\mathcal{Q}}_{j-1} & Q_j\end{pmatrix}.$$

Note that $\mathscr{L}_j$ can be computed without Q_j and that $V_{j+1}^*\widetilde{\mathcal{Q}}_{j-1}$ is used to compute $\underline{\mathscr{L}}_j$ and $\widetilde{\mathcal{Q}}_j$.

Proof. From (26)–(28), we have

$$\mathscr{L}_j = \mathscr{V}_j^*\begin{pmatrix}\mathscr{V}_j\underline{\mathscr{L}}_{j-1} + \widetilde{\mathcal{Q}}_{j-1} & AV_j\end{pmatrix} = \begin{pmatrix} & H_{1,j}\\ \underline{\mathscr{L}}_{j-1} & \vdots\\ & H_{j,j}\end{pmatrix},$$

$$\underline{\mathscr{L}}_j = \mathscr{V}_{j+1}^*\begin{pmatrix}\mathscr{V}_j\underline{\mathscr{L}}_{j-1} + \widetilde{\mathcal{Q}}_{j-1} & AV_j\end{pmatrix} = \begin{pmatrix}\mathscr{L}_j\\ V_{j+1}^*\widetilde{\mathcal{Q}}_{j-1}H_{j+1,j}\end{pmatrix}.$$

On the other hand

$$\widetilde{\mathcal{Q}}_j = (I - \mathscr{V}_{j+1}\mathscr{V}_{j+1}^*)\mathcal{Q}_j = \begin{pmatrix}(I - V_{j+1}V_{j+1}^*)\widetilde{\mathcal{Q}}_{j-1} & Q_j\end{pmatrix}$$

because $\mathscr{V}_j^*\mathcal{Q}_{j-1} = 0$ and $\mathscr{V}_{j+1}^*Q_j = 0$. □

3.2. BGMRES and inexact breakdowns

The introduction of Q_j in Algorithm 1 modifies the solution and residual of BGMRES. From (27), the solution is given by $X_j = \mathscr{V}_jY_j$ where Y_j solves

$$(\mathscr{V}_{j+1}\underline{\mathscr{L}}_j + \widetilde{\mathcal{Q}}_j)^*V_1\Lambda_1 = (\mathscr{V}_{j+1}\underline{\mathscr{L}}_j + \widetilde{\mathcal{Q}}_j)^*(\mathscr{V}_{j+1}\underline{\mathscr{L}}_j + \widetilde{\mathcal{Q}}_j)Y_j \tag{34}$$

or equivalently

$$\underline{\mathscr{L}}_j^* \underline{\Lambda}_j = (\underline{\mathscr{L}}_j^* \underline{\mathscr{L}}_j + \widetilde{\mathscr{Q}}_j^* \widetilde{\mathscr{Q}}_j) Y_j \quad \text{with } \underline{\Lambda}_j = \begin{pmatrix} \Lambda_j \\ 0 \end{pmatrix}. \tag{35}$$

From Proposition 1, we have

$$\underline{\mathscr{L}}_j^* \underline{\mathscr{L}}_j = \mathscr{L}_j^* \mathscr{L}_j + \begin{pmatrix} \widetilde{\mathscr{Q}}_{j-1}^* V_{j+1} V_{j+1}^* \widetilde{\mathscr{Q}}_{j-1} & \widetilde{\mathscr{Q}}_{j-1}^* V_{j+1} H_{j+1,j} \\ H_{j+1,j}^* V_{j+1}^* \widetilde{\mathscr{Q}}_{j-1} & H_{j+1,j}^* H_{j+1,j} \end{pmatrix},$$

$$\widetilde{\mathscr{Q}}_j^* \widetilde{\mathscr{Q}}_j = \begin{pmatrix} \widetilde{\mathscr{Q}}_{j-1}^* (I - V_{j+1} V_{j+1}^*) \widetilde{\mathscr{Q}}_{j-1} & \widetilde{\mathscr{Q}}_{j-1}^* Q_j \\ Q_j^* \widetilde{\mathscr{Q}}_{j-1} & Q_j^* Q_j \end{pmatrix}.$$

Now using (20), the system (35) reads as

$$\left(\mathscr{L}_j^* \mathscr{L}_j + \begin{pmatrix} \widetilde{\mathscr{Q}}_{j-1}^* \widetilde{\mathscr{Q}}_{j-1} & \widetilde{\mathscr{Q}}_{j-1}^* W_j \\ W_j^* \widetilde{\mathscr{Q}}_{j-1} & W_j^* W_j \end{pmatrix} \right) Y_j = \underline{\mathscr{L}}_j^* \underline{\Lambda}_j = \mathscr{L}_j^* \Lambda_j. \tag{36}$$

The residual R_j associated with X_j is given by

$$\begin{aligned} R_j &= V_1 \Lambda_1 - \left(\mathscr{V}_{j+1} \underline{\mathscr{L}}_j + \widetilde{\mathscr{Q}}_j \right) Y_j = \mathscr{V}_{j+1} \left(\underline{\Lambda}_j - \underline{\mathscr{L}}_j Y_j \right) - \widetilde{\mathscr{Q}}_j Y_j \\ &= \mathscr{V}_j (\Lambda_j - \mathscr{L}_j Y_j) - \begin{pmatrix} \widetilde{\mathscr{Q}}_{j-1} & W_j \end{pmatrix} Y_j. \end{aligned} \tag{37}$$

4. Detection of inexact breakdowns

We use the discussion and results of Section 3 to justify the criteria used to detect inexact breakdowns. We will say that $p_1 - p_{j+1} > 0$ inexact breakdowns occur at iteration j for the W-criterion when

$$\begin{aligned} \sigma_1 \begin{pmatrix} \widetilde{\mathscr{Q}}_{j-1} & W_j \end{pmatrix} \geqslant \cdots \geqslant \sigma_{p_{j+1}} \begin{pmatrix} \widetilde{\mathscr{Q}}_{j-1} & W_j \end{pmatrix} \geqslant \epsilon_j^{(W)} > \sigma_{p_{j+1}+1} \begin{pmatrix} \widetilde{\mathscr{Q}}_{j-1} & W_j \end{pmatrix} \\ \geqslant \cdots \geqslant \sigma_{p_1} \begin{pmatrix} \widetilde{\mathscr{Q}}_{j-1} & W_j \end{pmatrix} \end{aligned} \tag{38}$$

with the convention $\sigma_0 \begin{pmatrix} \widetilde{\mathscr{Q}}_{j-1} & W_j \end{pmatrix} = \epsilon_j^{(W)}$ where $\epsilon_j^{(W)}$ is a threshold parameter.

The R-criterion is defined in a similar way with the residual:

$$\sigma_{p_{j+1}}(R_j) \geqslant \epsilon_j^{(R)} > \sigma_{p_{j+1}+1}(R_j), \tag{39}$$

where $\epsilon_j^{(R)}$ is a threshold parameter.

These criteria can be seen as extensions of (11) and (19) to the inexact case. The inexact breakdowns detected by both criteria can be considered as sufficiently small perturbations of exact breakdowns to imply that a part of the exact solution is close enough to the Krylov space $\mathscr{K}_j(A, V_1)$. Of course, the quality of the computed solution depends on the choices of $\epsilon_j^{(W)}$ and $\epsilon_j^{(R)}$ that are discussed in Section 4.1.

Decomposition (20) and inclusion (24) can be accomplished with the Singular Value Decomposition algorithm (SVD) [4]. Note that only the computation of V_{j+1} satisfying (24) is required because $H_{j+1,j} = V_{j+1}^* W_j$ and $Q_j = W_j - V_{j+1} H_{j+1,j}$ are known from W_j and V_{j+1}.

For the W-criterion, the SVD of $\begin{pmatrix}\widetilde{\mathscr{Q}}_{j-1} & W_j\end{pmatrix}$ can formally be written

$$\begin{pmatrix}\widetilde{\mathscr{Q}}_{j-1} & W_j\end{pmatrix} = \mathbb{U}_1\Sigma_1\mathbb{V}_1^* + \mathbb{U}_2\Sigma_2\mathbb{V}_2^*, \tag{40}$$

where $\Sigma_1(\Sigma_2)$ contains the singular values of $\begin{pmatrix}\widetilde{\mathscr{Q}}_{j-1} & W_j\end{pmatrix}$ which are larger (smaller) than $\epsilon_j^{(W)}$. Then $V_{j+1} = \mathbb{U}_1$ satisfies (20) and (24).

The R-criterion does not allow a direct computation of V_{j+1} because in the SVD of

$$R_j = \mathbb{U}_1\Sigma_1\mathbb{V}_1^* + \mathbb{U}_2\Sigma_2\mathbb{V}_2^* \quad \text{with } \|\Sigma_2\|_2 < \epsilon_j^{(R)}, \tag{41}$$

$\mathbb{U}_1$ must be orthogonalized against $\mathscr{V}_j$ to obtain V_{j+1}. Practical details on these decompositions are given in Section 5.

4.1. Use of the W-criterion and the R-criterion

Suppose that BGMRES has converged at iteration s in the sense that

$$A\mathscr{V}_s = \mathscr{V}_s\mathscr{L}_s + \widetilde{\mathscr{Q}}_s.$$

In other words, $V_{s+1} = 0$. This means that $p_{s+1} = 0$ and (38) and (39) translate respectively into

$$\|\widetilde{\mathscr{Q}}_s\|_2 < \epsilon_s^{(W)} \quad \text{and} \quad \|R_s\|_2 < \epsilon_s^{(R)}. \tag{42}$$

As we have

$$R_s = B - AX_s = -\widetilde{\mathscr{Q}}_sY_s, \tag{43}$$

X_s is strongly connected to $\widetilde{\mathscr{Q}}_s$ and so its quality depends on the criterion used to compute $\widetilde{\mathscr{Q}}_s$. The properties of X_s will be described for each criterion. Moreover, since the method can be used to solve simultaneously several single linear systems, we will also study the quality of the approximate solution of

$$b = Ax, \quad \text{where } b = Bz \quad \text{with } z \in \mathbf{C}^p.$$

The analysis will allow to choose suitable values of $\epsilon_j^{(W)}$ and $\epsilon_j^{(R)}$ such that the final approximate solution X_s satisfies a requested condition.

4.1.1. Use of the W-criterion

From (42) and (43), the choice $\epsilon_j^{(W)} = \text{tol}$ for all $j = 1, \ldots, s$ ensures that

$$\frac{\|B - AX_s\|_2}{\|X_s\|_2} = \frac{\|\widetilde{\mathscr{Q}}_sY_s\|_2}{\|Y_s\|_2} \leqslant \|\widetilde{\mathscr{Q}}_s\|_2 < \text{tol} \tag{44}$$

and similarly for all $z \in \mathbf{C}^p$, $x_s = X_sz$ satisfies

$$\frac{\|b - Ax_s\|_2}{\|x_s\|_2} = \frac{\|\widetilde{\mathscr{Q}}_sY_sz\|_2}{\|Y_sz\|_2} < \text{tol} \quad \text{with } b = Bz. \tag{45}$$

These results show that the W-criterion allows to compute approximate solutions whose residual norms satisfy relative conditions in the block and single systems.

Moreover from Proposition 1 and (40), $\|\widetilde{\mathscr{Q}}_1\|_2 < \text{tol}$ and for $j \geqslant 2$, we have

$$\begin{aligned}\|\widetilde{\mathscr{Q}}_j\|_2 &= \left\|(I - V_{j+1}V_{j+1}^*)\begin{pmatrix}\widetilde{\mathscr{Q}}_{j-1} & Q_j\end{pmatrix}\right\|_2 = \left\|(I - V_{j+1}V_{j+1}^*)\begin{pmatrix}\widetilde{\mathscr{Q}}_{j-1} & W_j\end{pmatrix}\right\|_2 \\ &= \|\mathbb{U}_2\Sigma_2\mathbb{V}_2^*\|_2 < \text{tol}\end{aligned} \tag{46}$$

from which we see that for all $j \geqslant 1$ there are at most p_j singular values of $\begin{pmatrix}\widetilde{\mathscr{Q}}_{j-1} & W_j\end{pmatrix}$ larger or equal to tol. From (38) we see that this implies that $p_{j+1} \leqslant p_j$ and hence the sequence $\left(p_j\right)_{1 \leqslant j \leqslant s}$ is non-increasing.

The drawback of the W-criterion concerns the restart of BGMRES since (44) and (45) are no longer valid.

4.1.2. Use of the R-criterion

With the R-criterion and the choice $\epsilon_j^{(R)} = \text{tol}$ for all $j = 1, \ldots, s$, we have from (42)

$$\|B - AX_s\|_2 < \text{tol} \tag{47}$$

and for all $z \in \mathbf{C}^p$ with $\|z\|_2 = 1$, $x_s = X_s z$ satisfies

$$\|b - Ax_s\|_2 < \text{tol} \quad \text{with } b = Bz. \tag{48}$$

However no specific choice of $\epsilon_j^{(R)}$ allows to obtain a non-increasing sequence $(p_j)_{1 \leqslant j \leqslant s}$ since Y_j is used in the R-criterion and changes at each iteration. Note that the situation where $p_{j+1} > p_j$ means that $p_{j+1} - p_j$ vectors associated with inexact breakdowns at iteration $j - 1$ are no longer considered as inexact breakdowns at iteration j. These vectors are reintroduced in V_{j+1}.

An advantage of the R-criterion is that (47) and (48) remain satisfied when BGMRES is restarted. Moreover the R-criterion can be used to obtain relative conditions on the norm of the residual. Indeed, let $\widehat{X}_s$ be the approximate solution of

$$A\widehat{X} = V_1$$

computed by BGMRES with the R-criterion taking $\epsilon_j^{(R)} = \text{tol}$. Then $X_s = \widehat{X}_s \Lambda_1$ satisfies

$$\frac{\|B - AX_s\|_2}{\|B\|_2} = \frac{\|(V_1 - A\widehat{X}_s)\Lambda_1\|_2}{\|\Lambda_1\|_2} \leqslant \|V_1 - A\widehat{X}_s\|_2 < \text{tol}$$

and similarly for all $z \in \mathbf{C}^p$, we have

$$\frac{\|b - Ax_s\|_2}{\|b\|_2} < \text{tol} \quad \text{with } b = Bz \text{ and } x_s = X_s z.$$

4.1.3. Comparisons between the W- and R-criteria

The two previous subsections show that both criteria allow to compute approximate solutions with relative errors on the residual norms, and that the R-criterion preserves the conditions (47) and (48) during restarts, which is not the case with the W-criterion, i.e. the conditions (44) and (45) are altered during restarts when the W-criterion is used.

Now, one may ask if inexact breakdowns can be detected early in the iterative process. The answer is yes if the R-criterion is used since the condition (48) clearly implies that p_1 inexact breakdowns have been detected. To check if the W-criterion possesses such a property, we first need the following proposition.

Proposition 2. *The condition* (45) *is equivalent to*

$$\|\Gamma_s\|_2 < \text{tol}, \quad \textit{where } \Gamma_s = (B - AX_s)F_s \quad \textit{with } F_s = (X_s^* X_s)^{-1/2}. \tag{49}$$

Proof. Note first that since $\underline{\mathscr{H}}_s$ is non-singular, the matrix X_s is full rank and F_s is well defined.

Assume that (45) is satisfied and take $z = F_s u$ where u is the right singular vector associated with the singular value $\|\Gamma_s\|_2$. Then

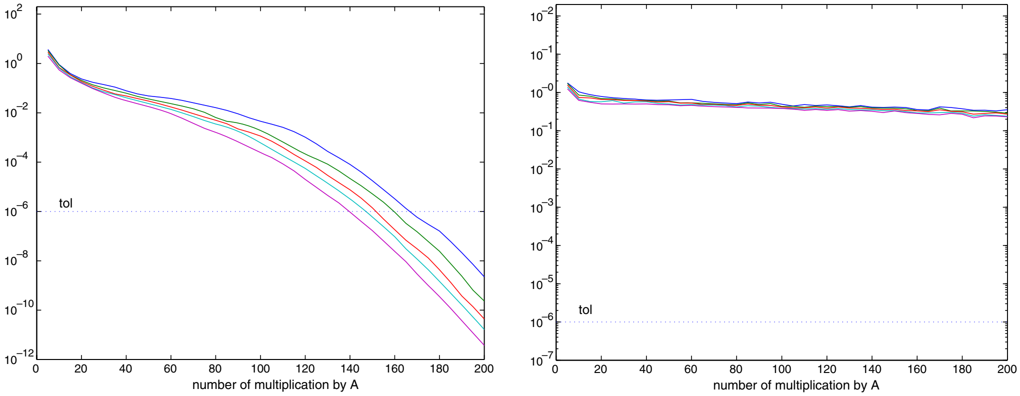

Fig. 1. Singular values of Γ_j (left) and $(\widetilde{\mathscr{Q}}_{j-1} \quad W_j)$ (right).

$$\text{tol} > \frac{\|(B - AX_s)z\|_2}{\|X_s z\|_2} = \frac{\|\Gamma_s u\|_2}{\|u\|_2} = \|\Gamma_s\|_2.$$

Conversely if (49) is satisfied then for all $z \in \mathbf{C}^p$, we have

$$\|(B - AX_s)z\|_2 \leqslant \|\Gamma_s\|_2 \|(X_s^* X_s)^{1/2} z\|_2 < \text{tol}\|X_s z\|_2,$$

which shows that (45) (and also (44)) are satisfied. □

Remark. The condition (44) does not necessarily imply (49).

The numerical test reported in Section 6, Fig. 1 shows that the condition (49) can be satisfied though the W-criterion detects no inexact breakdowns.

The following argument helps to understand why the W-criterion may be inefficient to detect inexact breakdowns.

Proposition 3. *Assume that BGMRES is used with the W-criterion and that at iterations $j-1$ and j the approximate solutions $X_{j-1} = \mathscr{V}_{j-1} Y_{j-1}$ and $X_j = \mathscr{V}_j Y_j$ satisfy*

$$\|X - X_j\|_2 \leqslant \epsilon \quad \textit{and} \quad \|X - X_{j-1}\| \leqslant (1+\nu)\epsilon$$

with ϵ small enough, $0 < \epsilon \ll \sigma_{\min}(X)$ and $0 < \nu \ll 1$.

Let $Y_j^{(1)} \in \mathbf{C}^{n_{j-1} \times p_1}$ and $Y_j^{(2)} \in \mathbf{C}^{p_j \times p_1}$ be such that $Y_j = \begin{pmatrix} Y_j^{(1)} \\ Y_j^{(2)} \end{pmatrix}$. Then

$$| \|Y_j^{(1)}\|_2 - \|X\|_2 | \leqslant (3 + 2\nu)\epsilon, \tag{50}$$

$$\|Y_j^{(2)}\|_2 \leqslant (2+\nu)\epsilon, \tag{51}$$

$$\|\widetilde{\mathscr{Q}}_{j-1} Y_j^{(1)} + W_j Y_j^{(2)}\|_2 \leqslant \|R_j\|_2 \leqslant \epsilon \|A\|_2, \tag{52}$$

$$\|\Gamma_j\|_2 \leqslant \epsilon \frac{\|A\|_2}{\sigma_{\min}(X) - \epsilon}. \tag{53}$$

Proof. On one hand we have

$$\|X_j - X_{j-1}\|_2^2 = \|\mathscr{V}_{j-1}(Y_j^{(1)} - Y_{j-1})\|_2^2 + \|V_{j+1} Y_j^{(2)}\|_2^2.$$

On the other hand

$$\|X_j - X_{j-1}\|_2 \leqslant \|X_j - X\|_2 + \|X - X_{j-1}\|_2 \leqslant (2+\nu)\epsilon.$$

Hence

$$\|Y_j^{(1)} - Y_{j-1}\|_2 \leqslant (2+\nu)\epsilon \quad \text{and} \quad \|Y_j^{(2)}\|_2 \leqslant (2+\nu)\epsilon,$$

and

$$\begin{aligned} |\|Y_j^{(1)}\|_2 - \|X\|_2| &\leqslant \|\mathscr{V}_{j-1}Y_j^{(1)} - X\|_2 \\ &\leqslant \|Y_j^{(1)} - Y_{j-1}\|_2 + \|X_{j-1} - X\|_2 \leqslant (3+2\nu)\epsilon. \end{aligned}$$

Moreover $\|R_j\|_2$, $\|F_j\|_2$ and $\|\Gamma_j\|_2$ are bounded as follows:

$$\begin{aligned} \|R_j\|_2 &= \|A(X - X_j)\|_2 \leqslant \epsilon\|A\|_2 \\ \|F_j\|_2^{-1} &= \sigma_{\min}(X_j) \geqslant \sigma_{\min}(X) - \|X_j - X\|_2 \geqslant q\sigma_{\min}(X) - \epsilon > 0 \\ \|\Gamma_j\|_2 &\leqslant \|R_j\|_2\|F_j\|_2 \leqslant \frac{\epsilon\|A\|_2}{\sigma_{\min}(X) - \epsilon}. \end{aligned}$$

From (37), we also have

$$\|R_j\|_2^2 = \|\mathscr{V}_j(\Lambda_j - \mathscr{L}_jY_j)\|_2^2 + \left\|\begin{pmatrix}\widetilde{\mathscr{Q}}_{j-1} & W_j\end{pmatrix} Y_j\right\|_2^2 \geqslant \|\widetilde{\mathscr{Q}}_{j-1}Y_j^{(1)} + W_jY_j^{(2)}\|_2^2. \qquad \square$$

From (46), (50) and (51) we see that $\|\widetilde{\mathscr{Q}}_{j-1}\|_2$ is small, $\|Y_j^{(1)}\|_2 \approx \|X\|_2$ and $\|Y_j^{(2)}\|_2$ is small. Therefore (52) shows that the smallness of $\|R_j\|_2$ does not necessarily result from $\|W_j\|_2$. In other words, even if $\|R_j\|_2$ and $\|\Gamma_j\|_2$ are small, $\|\begin{pmatrix}\widetilde{\mathscr{Q}}_{j-1} & W_j\end{pmatrix}\|_2$ may be large.

In conclusion, the W-criterion is not sharp because it is based on a strong condition implying that inexact breakdowns may be detected too late and thus leading to useless computations.

5. Implementation details and algorithms

We have seen in Proposition 1 that $\underline{\mathscr{L}}_j$ and $\widetilde{\mathscr{Q}}_j$ can be updated at each iteration j. Moreover, the solution and residual associated to BGMRES can be computed before the decomposition (20) (see (36) and (37)). These remarks will be used to derive an economical version of BGMRES.

From Theorem 2, there exist P_j and G_j such that

$$P_jG_j = \widetilde{\mathscr{Q}}_j \quad \text{with} \begin{cases} P_j \in \mathbf{C}^{N\times\tilde{q}_j} \text{ has orthonormal columns with } \mathscr{V}_{j+1}^*P_j = 0, \\ G_j \in \mathbf{C}^{\tilde{q}_j\times n_j} \text{ full rank,} \end{cases} \tag{54}$$

where $\tilde{q}_j = \operatorname{rank}(\widetilde{\mathscr{Q}}_j) \leqslant p_1 - p_{j+1}$. Then the matrices W_j and P_{j-1} can be stored in a block of at most p_1 columns. Note that $\tilde{q}_j < p_1 - p_{j+1}$ only when exact breakdowns occur. In practice, we set $\tilde{q}_j = p_1 - p_{j+1}$ to simplify the computations of P_j and G_j.

We now discuss some important details in the implementation aspect. In particular we explain how to compute the SVD of R_j and how to compute the last block row of $\mathscr{L}_{j+1}$ denoted by $\mathscr{L}_{j+1,:}$ and the matrices V_{j+1}, P_j and G_j in an economical way. It is important to notice that the product P_jG_j is never formed explicitly. We first recall the following facts:

- The matrix V_{j+1} is obtained from the SVD of $\begin{pmatrix}P_{j-1}G_{j-1} & W_j\end{pmatrix}$ or R_j (see (40) and (41)).

- From Proposition 1, we have the formulas:

$$\begin{aligned}\mathscr{L}_{j+1,:} &= \begin{pmatrix} V_{j+1}^* P_{j-1} G_{j-1} & H_{j+1,j} \end{pmatrix} = \begin{pmatrix} V_{j+1}^* P_{j-1} G_{j-1} & V_{j+1}^* W_j \end{pmatrix} \\ &= V_{j+1}^* \begin{pmatrix} P_{j-1} G_{j-1} & W_j \end{pmatrix},\end{aligned} \tag{55}$$

$$\begin{aligned}P_j G_j &= \begin{pmatrix} (I - V_{j+1} V_{j+1}^*) P_{j-1} G_{j-1} & Q_j \end{pmatrix} = (I - V_{j+1} V_{j+1}^*) \begin{pmatrix} P_{j-1} G_{j-1} & W_j \end{pmatrix} \\ &= \begin{pmatrix} P_{j-1} G_{j-1} & W_j \end{pmatrix} - V_{j+1} \mathscr{L}_{j+1,:}.\end{aligned} \tag{56}$$

In practice, we proceed as follows:

- Let $C_j = P_{j-1}^* W_j$ and denote by $\widetilde{W}_j D_j$ the reduced QR factorization of $W_j - P_{j-1} C_j$. Then $\begin{pmatrix} P_{j-1} & \widetilde{W}_j \end{pmatrix}$ has orthonormal columns and

$$\begin{pmatrix} P_{j-1} G_{j-1} & W_j \end{pmatrix} = \begin{pmatrix} P_{j-1} & \widetilde{W}_j \end{pmatrix} \begin{pmatrix} G_{j-1} & C_j \\ 0 & D_j \end{pmatrix}. \tag{57}$$

- With the W-criterion, the SVD of $\begin{pmatrix} P_{j-1} G_{j-1} & W_j \end{pmatrix}$ is obtained from that of

$$\begin{pmatrix} G_{j-1} & C_j \\ 0 & D_j \end{pmatrix} = \mathbb{U}_1 \Sigma_1 \mathbb{V}_1^* + \mathbb{U}_2 \Sigma_2 \mathbb{V}_2^*$$

as

$$\begin{pmatrix} P_{j-1} G_{j-1} & W_j \end{pmatrix} = \begin{pmatrix} P_{j-1} & \widetilde{W}_j \end{pmatrix} \mathbb{U}_1 \Sigma_1 \mathbb{V}_1^* + \begin{pmatrix} P_{j-1} & \widetilde{W}_j \end{pmatrix} \mathbb{U}_2 \Sigma_2 \mathbb{V}_2^*.$$

Then from (55) and (56), we have $V_{j+1} = \begin{pmatrix} P_{j-1} & \widetilde{W}_j \end{pmatrix} \mathbb{U}_1$, $\mathscr{L}_{j+1,:} = \Sigma_1 \mathbb{V}_1^*$, $P_j = \begin{pmatrix} P_{j-1} & \widetilde{W}_j \end{pmatrix} \mathbb{U}_2$ and $G_j = \Sigma_2 \mathbb{V}_2^*$.

- With the R-criterion, the computation of V_{j+1} requires much more manipulations. From (37) and (57), we have

$$\begin{aligned}R_j &= \mathscr{V}_j (\Lambda_j - \mathscr{L}_j Y_j) - \begin{pmatrix} P_{j-1} & \widetilde{W}_j \end{pmatrix} \begin{pmatrix} G_{j-1} & C_j \\ 0 & D_j \end{pmatrix} Y_j \\ &= \begin{pmatrix} \mathscr{V}_j & \begin{pmatrix} P_{j-1} & \widetilde{W}_j \end{pmatrix} \end{pmatrix} \begin{pmatrix} \Lambda_j - \mathscr{L}_j Y_j \\ -\begin{pmatrix} G_{j-1} & C_j \\ 0 & D_j \end{pmatrix} Y_j \end{pmatrix}.\end{aligned} \tag{58}$$

Since the columns of $\begin{pmatrix} \mathscr{V}_j & \begin{pmatrix} P_{j-1} & \widetilde{W}_j \end{pmatrix} \end{pmatrix}$ are orthonormal, we obtain the SVD of R_j

$$R_j = \begin{pmatrix} \mathscr{V}_j & \begin{pmatrix} P_{j-1} & \widetilde{W}_j \end{pmatrix} \end{pmatrix} \mathbb{U}_1 \Sigma_1 \mathbb{V}_1^* + \begin{pmatrix} \mathscr{V}_j & \begin{pmatrix} P_{j-1} & \widetilde{W}_j \end{pmatrix} \end{pmatrix} \mathbb{U}_2 \Sigma_2 \mathbb{V}_2^*$$

from that of

$$\begin{pmatrix} \Lambda_j - \mathscr{L}_j Y_j \\ -\begin{pmatrix} G_{j-1} & C_j \\ 0 & D_j \end{pmatrix} Y_j \end{pmatrix} = \mathbb{U}_1 \Sigma_1 \mathbb{V}_1^* + \mathbb{U}_2 \Sigma_2 \mathbb{V}_2^*.$$

To compute V_{j+1} satisfying (20), we must orthogonalize $\begin{pmatrix} \mathscr{V}_j & \begin{pmatrix} P_{j-1} & \widetilde{W}_j \end{pmatrix} \end{pmatrix}$ against $\mathscr{V}_j$. If we partition $\mathbb{U}_1 = \begin{pmatrix} \mathbb{U}_1^{(1)} \\ \mathbb{U}_1^{(2)} \end{pmatrix}$ in accordance with $\begin{pmatrix} \mathscr{V}_j & \begin{pmatrix} P_{j-1} & \widetilde{W}_j \end{pmatrix} \end{pmatrix}$, then

$$\begin{aligned}&(I - \mathscr{V}_j \mathscr{V}_j^*) \begin{pmatrix} \mathscr{V}_j & \begin{pmatrix} P_{j-1} & \widetilde{W}_j \end{pmatrix} \end{pmatrix} \mathbb{U}_1 \\ &\quad = \begin{pmatrix} 0 & \begin{pmatrix} P_{j-1} & \widetilde{W}_j \end{pmatrix} \end{pmatrix} \begin{pmatrix} \mathbb{U}_1^{(1)} \\ \mathbb{U}_1^{(2)} \end{pmatrix} = \begin{pmatrix} P_{j-1} & \widetilde{W}_j \end{pmatrix} \mathbb{U}_1^{(2)}.\end{aligned}$$

Now, let $\mathbb{W}_1$ and $\mathbb{W}_2$ be such that

$$\mathrm{Range}\,\mathbb{W}_1 = \mathrm{Range}\,\mathbb{U}_1^{(2)} \quad \text{and} \quad \begin{pmatrix}\mathbb{W}_1 & \mathbb{W}_2\end{pmatrix} \text{ is unitary.}$$

Then from (55) and (56), we have

$$\begin{aligned}
V_{j+1} &= \begin{pmatrix}P_{j-1} & \widetilde{W}_j\end{pmatrix}\mathbb{W}_1,\\
\mathscr{L}_{j+1,:} &= \mathbb{W}_1^* \begin{pmatrix}G_{j-1} & C_j\\ 0 & D_j\end{pmatrix},\\
P_j &= \begin{pmatrix}P_{j-1} & \widetilde{W}_j\end{pmatrix}\mathbb{W}_2,\\
G_j &= \mathbb{W}_2^* \begin{pmatrix}G_{j-1} & C_j\\ 0 & D_j\end{pmatrix}.
\end{aligned}$$

The solution Y_j of the least squares problem (14) can be computed using the normal equations (36) or, preferably, using (26) and (57)

$$\begin{aligned}
A\mathscr{V}_j &= \mathscr{V}_j\mathscr{L}_j + \begin{pmatrix}P_{j-1} & \widetilde{W}_j\end{pmatrix}\begin{pmatrix}G_{j-1} & C_j\\ 0 & D_j\end{pmatrix}\\
&= \begin{pmatrix}\mathscr{V}_j & \begin{pmatrix}P_{j-1} & \widetilde{W}_j\end{pmatrix}\end{pmatrix}\begin{pmatrix}\mathscr{L}_j\\ \begin{pmatrix}G_{j-1} & C_j\\ 0 & D_j\end{pmatrix}\end{pmatrix}.
\end{aligned}$$

Since the columns of $\begin{pmatrix}\mathscr{V}_j & \begin{pmatrix}P_{j-1} & \widetilde{W}_j\end{pmatrix}\end{pmatrix}$ are orthonormal, the solution Y_j solves the linear squares problem

$$\min_{Y\in\mathbf{C}^{n_j\times p_1}} \left\| \begin{pmatrix}\mathscr{L}_j\\ \begin{pmatrix}G_{j-1} & C_j\\ 0 & D_j\end{pmatrix}\end{pmatrix} Y - \underline{\Lambda}_j \right\|_2 .$$

We summarize the discussion above in the following algorithm.

Algorithm 2 (*BGMRES with inexact breakdowns*)

Let $B = V_1\Lambda_1$, $P_0 = 0$, $G_0 = 0$ and $\underline{\mathscr{L}}_0 = [\ ]$.

For $j = 1, \ldots, m$

1. Compute $\mathscr{L}_{1,1:j}$ and W_j as

$$\begin{aligned}
W_j &= AV_j,\\
\mathscr{L}_{1,1:j} &= \mathscr{V}_j^* W_j,\\
W_j &= W_j - \mathscr{V}_j\mathscr{L}_{1,1:j},
\end{aligned}$$

   where $\mathscr{V}_j = \begin{pmatrix}V_1 & \cdots & V_j\end{pmatrix}$.
2. Set

$$\mathscr{L}_j = \begin{pmatrix}\underline{\mathscr{L}}_{j-1} & \mathscr{L}_{1,1:j}\end{pmatrix}.$$

3. Compute C_j, $\widetilde{W}_j$ and D_j such that

$$\begin{aligned}
C_j &= P_{j-1}^* W_j,\\
\widetilde{W}_j D_j &= W_j - P_{j-1}C_j \quad \text{(reduced QR factorization).}
\end{aligned}$$

4. The W-criterion:
   - Compute the SVD

$$\begin{pmatrix} G_{j-1} & C_j \\ 0 & D_j \end{pmatrix} = \mathbb{U}_1 \Sigma_1 \mathbb{V}_1^* + \mathbb{U}_2 \Sigma_2 \mathbb{V}_2^*, \quad \text{where } \sigma_{\min}(\Sigma_1) \geqslant \epsilon_j^{(W)} > \|\Sigma_2\|_2.$$

   - Compute V_{j+1}, $\mathscr{L}_{j+1,:}$, P_j and G_j as

$$\begin{aligned} V_{j+1} &= \begin{pmatrix} P_{j-1} & \widetilde{W}_j \end{pmatrix} \mathbb{U}_1, \\ \mathscr{L}_{j+1,:} &= \Sigma_1 \mathbb{V}_1^*, \\ P_j &= \begin{pmatrix} P_{j-1} & \widetilde{W}_j \end{pmatrix} \mathbb{U}_2, \\ G_j &= \Sigma_2 \mathbb{V}_2^*. \end{aligned}$$

5. The R-criterion:
   - Compute Y_j the solution of

$$\min_{Y \in \mathbf{C}^{n_j \times p_1}} \left\| \begin{pmatrix} \mathscr{L}_j \\ \begin{pmatrix} G_{j-1} & C_j \\ 0 & D_j \end{pmatrix} \end{pmatrix} Y - \underline{\Lambda}_j \right\|_2 \quad \text{with } \underline{\Lambda}_j = \begin{pmatrix} \Lambda_1 \\ 0 \end{pmatrix}$$

   - Compute the SVD

$$\begin{pmatrix} \Lambda_j - \mathscr{L}_j Y_j \\ -\begin{pmatrix} G_{j-1} & C_j \\ 0 & D_j \end{pmatrix} Y_j \end{pmatrix} = \mathbb{U}_1 \Sigma_1 \mathbb{V}_1^* + \mathbb{U}_2 \Sigma_2 \mathbb{V}_2^*, \quad \text{where } \sigma_{\min}(\Sigma_1) \geqslant \epsilon_j^{(R)} > \|\Sigma_2\|_2.$$

   - Compute $\mathbb{W}_1$ and $\mathbb{W}_2$ such that

$$\mathrm{Range}\,\mathbb{W}_1 = \mathrm{Range}\,\mathbb{U}_1^{(2)} \quad \text{with } \mathbb{U}_1 = \begin{pmatrix} \mathbb{U}_1^{(1)} \\ \mathbb{U}_1^{(2)} \end{pmatrix} \quad \text{and } \begin{pmatrix} \mathbb{W}_1 & \mathbb{W}_2 \end{pmatrix} \text{ is unitary.}$$

   - Compute V_{j+1}, $\mathscr{L}_{j+1,:}$, P_j and G_j as

$$\begin{aligned} V_{j+1} &= \begin{pmatrix} P_{j-1} & \widetilde{W}_j \end{pmatrix} \mathbb{W}_1, \\ \mathscr{L}_{j+1,:} &= \mathbb{W}_1^* \begin{pmatrix} G_{j-1} & C_j \\ 0 & D_j \end{pmatrix}, \\ P_j &= \begin{pmatrix} P_{j-1} & \widetilde{W}_j \end{pmatrix} \mathbb{W}_2, \\ G_j &= \mathbb{W}_2^* \begin{pmatrix} G_{j-1} & C_j \\ 0 & D_j \end{pmatrix}. \end{aligned}$$

6. Set

$$\underline{\mathscr{L}}_j = \begin{pmatrix} \mathscr{L}_j \\ \mathscr{L}_{j+1,:} \end{pmatrix}.$$

End For j.

Next, we compare the computational cost involved in Algorithm 2 and in the classical methods based on Algorithm 1. To simplify the comparison, we only compare the number of operations among large vectors per iteration. For Algorithm 1, we assume that no exact breakdown has occurred and so the size of V_j is always equal to p_1 and the size of $\mathscr{V}_j$ denoted by $n_j^{(1)}$ equals

Table 1
Operations with large vectors at step j of Algorithms 1 and 2

Algorithm 1	Flops	Algorithm 2	Flops
$\mathscr{V}_j^* W_j$	$2n_j^{(1)} p_1 N$	$\mathscr{V}_j^* W_j$	$2n_j^{(2)} p_j N$
$\mathscr{V}_j(\mathscr{V}_j^* W_j)$	$2n_j^{(1)} p_1 N$	$\mathscr{V}_j(\mathscr{V}_j^* W_j)$	$2n_j^{(2)} p_j N$
$W_j - \mathscr{V}_j(\mathscr{V}_j^* W_j)$	$p_1 N$	$W_j - \mathscr{V}_j(\mathscr{V}_j^* W_j)$	$p_j N$
QR factorization	$2p_1^2 N$	$P_{j-1}^* W_j$	$2(p_1 - p_j) p_j N$
$V_{j+1} H_{j+1,j} = W_j$		$P_{j-1}(P_{j-1}^* W_j)$	$2(p_1 - p_j) p_j N$
		$W_j - P_{j-1}(P_{j-1}^* W_j)$	$p_j N$
		QR factorization	$p_j^2 N$
		$\widetilde{W}_j D_j = W_j - P_{j-1} C_j$	
		$\left(P_{j-1} \quad \widetilde{W}_j\right) \mathbb{U}_1$ or $\left(P_{j-1} \quad \widetilde{W}_j\right) \mathbb{W}_1$	$2p_{j+1} p_1 N$
		$\left(P_{j-1} \quad \widetilde{W}_j\right) \mathbb{U}_2$ or $\left(P_{j-1} \quad \widetilde{W}_j\right) \mathbb{W}_2$	$2(p_1 - p_{j+1}) p_1 N$
Total: $T_j^{(1)} = (4n_j^{(1)} + 2p_1 + 1) p_1 N$		$T_j^{(2)} = \left(p_j(4n_j^{(2)} + 4p_1 - 3p_j + 2) + 2p_1^2\right) N$	

jp_1. For Algorithm 2, we assume that $p_1 - p_j \geqslant 1$ and $p_1 - p_{j+1} \geqslant 1$ inexact breakdowns have occurred at iterations $j - 1$ and j and we denote by $n_j^{(2)}$ the size of $\mathscr{V}_j$. Table 1 shows the computational costs, $T_j^{(1)}$ and $T_j^{(2)}$, of one iteration j in Algorithms 1 and 2.

The cost of vector-multiplications by A is not included. From this table we see that T_2 decreases since $n_j^{(2)}$ decreases and that

$$T_j^{(1)} - T_j^{(2)} = 4(p_1 n_j^{(1)} - p_j n_j^{(2)})N - (p_j(4p_1 - 3p_j + 2) - p_1)N.$$

As the iterations unfold, p_j decreases and $n_j^{(1)} - n_j^{(2)}$ increases. Therefore, the cost of Algorithm 2 becomes smaller than that of Algorithm 1. Recall that each iteration j of Algorithm 1 (Algorithm 2) necessitates p_1 ($p_j \leqslant p_1$) vector-multiplications by A. Fig. 3(right) shows the total number of flops $\sum_j T_j^{(1)}$ and $\sum_j T_j^{(2)}$ required for BGMRES using Algorithms 1 and 2.

6. Numerical illustrations

We illustrate the behavior of Algorithm 2 with the W- and especially the R-criteria. We also compare with BGMRES when the vectors associated with inexact breakdowns are deleted. We consider the linear system

$$AX = B \quad \text{with } A = \widetilde{A}M, \tag{59}$$

where M is an approximation of $\widetilde{A}^{-1}$ obtained with Matlab incomplete LU factorization with 0 level of fill-in. A is the matrix SHERMAN5 of size 3312 taken from the Harwell–Boeing collection of test matrices[1]. The right-hand side B has p orthonormal columns generated with normal distribution. When the R-criterion is used, Algorithm 2 is restarted when the size of block Krylov space reaches a maximum denoted by n and is stopped when p inexact breakdowns have occurred.

[1] http://math.nist.gov/MatrixMarket/

6.1. Behavior of the W-criterion

The following tests are performed taking $\mathrm{tol} = 10^{-6}$ and $p = 5$. Recall that approximate solutions computed using the W-criterion do not satisfy (44) and (45) when BGMRES is restarted. Therefore we take $n = 200$ and no restart is used. Fig. 1 shows the singular values of Γ_j (see Proposition 2) and $\begin{pmatrix}\widetilde{\mathcal{Q}}_{j-1} & W_j\end{pmatrix}$ respectively with the W-criterion. We see that the W-criterion detects no breakdown. However before the iterations terminate, we have $\|\Gamma_j\|_2 \leqslant \mathrm{tol}$ and then X_j satisfies (44) and (45). In the figure, it clearly appears that the singular values of $\begin{pmatrix}\widetilde{\mathcal{Q}}_{j-1} & W_j\end{pmatrix}$ stay large whereas those of Γ_j decrease (see Proposition 3).

6.2. Behavior of the R-criterion

To illustrate the behavior of BGMRES with the R-criterion, we take $n = 75$. Unlike the tests with the W-criterion, here restarts are used. Fig. 2 shows the singular values of the residuals computed by the formula (37). The figure also shows that inexact breakdowns occur when the curve goes below the parameter tol. The R-criterion is clearly efficient to detect inexact breakdowns. When inexact breakdowns are detected (i.e. $p_j < p = 5$), the singular values of the residual have two types of convergence. First, the $p - p_j$ singular values smaller than tol stay almost unchanged. This means that no computation associated with them is performed. The speed of convergence of the other singular values increases since only p_j matrix-vector multiplies are done at each iteration. The ranks of the blocks $V_j's$ and the residual norms are also shown in Fig. 2.

6.3. Is handling inexact breakdowns worth the effort?

To answer this question, we need to compare the solutions obtained with the inexact breakdown strategy (BGMRES with the R-criterion) and the classical methods, i.e. BGMRES based on Algorithm 1 where the factorization $W_j = V_{j+1}H_{j+1,j}$ uses a standard reduced QR factorization. In Matlab notation $[V_{j+1}, H_{j+1,j}] = \mathrm{qr}(W_j, 0)$. Actually this version amounts to using Algorithm 2 with the W-criterion setting $P_j = 0$ and $G_j = 0$ for all j and choosing $\Sigma_2 = 0$. The comparisons shown in Table 1 and Fig. 3(right) partially answer this question: the computational cost decreases

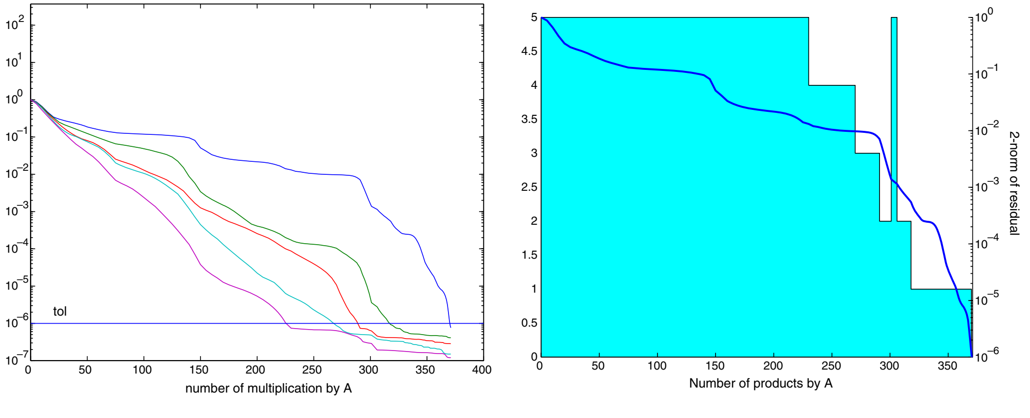

Fig. 2. BGMRES with the R-criterion: singular values of R_j (left), block sizes and residual norms with the R-criterion (right).

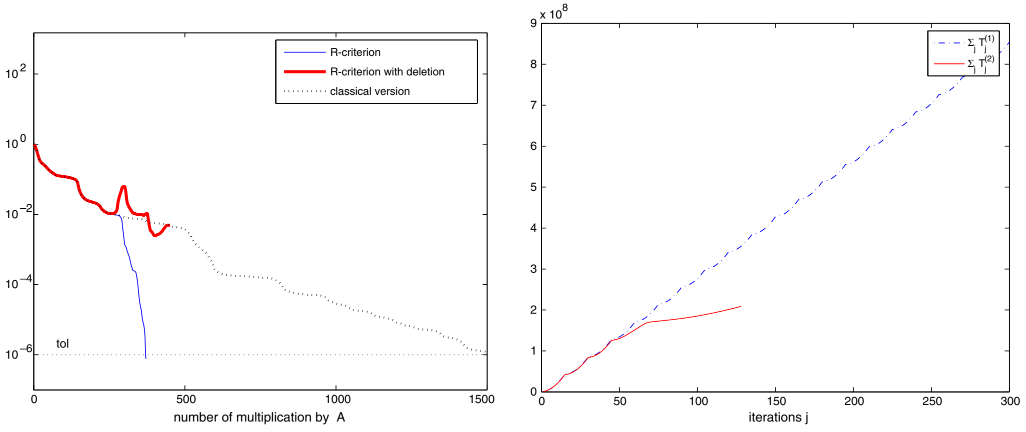

Fig. 3. BGMRES with and without the R-criterion: residual norms (left), number of flops (right).

when inexact breakdowns are taken into account. The numerical tests confirm the advantage of inexact breakdown strategies over the classical methods, see Fig. 3(left).

We also compare with another strategy where the matrices P_j and G_j are deleted in the R-criterion. More precisely, in Algorithm 2, we use the R-criterion and set $P_j = 0$ and $G_j = 0$ for all j. We briefly analyze this strategy in Section 6.4.

6.4. *Deletion of inexact breakdowns*

Assume that no inexact breakdown has been detected before iteration j. Then

$$A\mathscr{V}_j = \mathscr{V}_j\mathscr{H}_j + \begin{pmatrix}0 & W_j\end{pmatrix}.$$

If, as in (20) and (24), W_j is decomposed in form $W_j = V_{j+1}H_{j+1,j} + Q_j$ with $V_{j+1}^*V_{j+1} = I$ and $\operatorname{Range} V_{j+1} \subset \operatorname{Range} W_j$, then we have

$$A_j\mathscr{V}_j = \mathscr{V}_{j+1}\underline{\mathscr{H}}_j \quad \text{with } A_j = A - \begin{pmatrix}0 & Q_j\end{pmatrix}\mathscr{V}_j^* \equiv A - \mathscr{Q}_j\mathscr{V}_j^*.$$

This process can be continued with A replaced by A_j. Since the rank of V_{j+1} is decreasing, we will have at the last iteration s, $A_s\mathscr{V}_s = \mathscr{H}_s\mathscr{V}_s$ and then

$$A_sX_s = B \quad \text{with } A_s = A - \mathscr{Q}_s\mathscr{V}_s^*.$$

Clearly an advantage here is the low computational cost since $P_jG_j = \mathscr{Q}_j$ is not used at iteration j of the process. We are naturally interested in comparing the solutions obtained with and without deletion of inexact breakdowns.

Note that, in general, at most p matrices Q_j are non-null during the whole process since at most, p inexact breakdowns can occur.

With the W-criterion, we have $\|Q_j\|_2 < \epsilon_j^{(W)}$ and assuming that $\epsilon_j^{(W)} = \mathrm{tol}/\sqrt{p}$ for all $1 \leqslant j \leqslant s$, we have

$$\|A - A_s\|_2 < \mathrm{tol}.$$

This shows that X_s satisfies (44) and (45). Therefore, with the W-criterion, the deletion of inexact breakdowns only requires to take a little smaller $\epsilon_j^{(W)}$.

With the R-criterion, the perturbations introduced by the deletion may be larger. Actually, at iteration j, the detection of inexact breakdowns is based on Y_j but at the last iteration, Y_j is replaced by Y_s since

$$\|B - AX_s\|_2 = \|\mathcal{Q}_s Y_s\|_2.$$

Algorithms 2 is tested with the R-criterion when inexact breakdowns are deleted using the same parameters as in the previous tests. Note that now the formula (37) is no longer valid for the residual. For this reason, the norm of the explicit residual $R_j^{(\mathrm{e})} = B - AX_j$ with and without deletion of inexact breakdowns is plotted in Fig. 3(left) where it appears that the deletion of inexact breakdowns may deteriorate the quality of the computed solution.

7. Conclusion

We have extended the notion of breakdown in a way appropriate to the block GMRES method (BGMRES) for solving linear systems with several right-hand sides. From a theoretical point of view, an exact breakdown occurs at iteration j when the rank of the matrix W_j constructed by the block Arnoldi algorithm (Algorithm 1) is less than that of the right hand side of the block linear system (1), or equivalently when the residual R_j associated to BGMRES is rank-deficient (see (11) and (19)). This is a favorable situation since it means that a part of the block linear system under consideration has been solved exactly. From a numerical point of view, the inexact breakdown is much more likely to occur. It gives rise to an approximate solution. The inexact breakdowns denoted by the matrix $\tilde{\mathcal{Q}}_j$ (see (20), (21), and (24)) correspond to the situation where some singular values of $(\tilde{\mathcal{Q}}_{j-1} \quad W_j)$ (W-criterion) or R_j (R-criterion) are smaller than some tolerance parameters. However, in this case, there is no correspondence between the singular values of $(\tilde{\mathcal{Q}}_{j-1} \quad W_j)$ and R_j.

We showed that the deletion of the vectors associated with inexact breakdowns is not recommended and analyzed and compared the W- and R-criteria in the framework of BGMRES. It follows that both criteria allow to compute approximate solutions with relative errors on the residual norms for the block and single systems. Unlike the W-criterion, the R-criterion can be used when BGMRES is restarted and detects efficiently the inexact breakdowns, thus allowing a decrease in the computational cost and memory requirements. However, with the R-criterion, the deletion of inexact breakdowns may deteriorate the solution. In conclusion, a version of BGMRES along the lines suggested in Algorithm 2, using the R-criterion, is advisable.

All the results apply to the block FOM method (see [11]).

Acknowledgments

The authors gratefully acknowledge the influence of the referees on this paper.

References

[1] W.E. Arnoldi, The principle of minimized iterations in the solution of the matrix eigenvalue problem, Quart. Appl. Math. 9 (1951) 17–29.

[2] R.W. Freund, N.M. Nachtigal, QMR: A quasi-minimal residual method for non-Hermitian linear systems, Numer. Math. 60 (1991) 315–339.

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

[4] G.H. Golub, C.F. Van Loan, Matrix Computations, second ed., The Johns Hopkins Press, Baltimore, MD, 1989.
[5] M. Gutknecht, Solvers for systems of equations with multiple right-hand sides, Talk given at GAMM workshop on Numerical Linear Algebra in Bielefeld, 2002, see http://www.sam.math.ethz.ch/mhg/.
[6] K. Jbilou, A. Messaoudi, H. Sadok, Global FOM and GMRES algorithms for matrix equations, Appl. Numer. Math. 32 (1999) 49–63.
[7] C. Lanczos, An iteration method for the solution of the eigenvalue problem of linear differential and integral operators, J. Res. Nat. Bur. Standards 45 (1950) 255–282.
[8] J. Langou, Iterative methods for solving linear systems with multiple right-hand sides, Ph.D. dissertation TH/PA/03/24, CERFACS, France, 2003.
[9] A.A. Nikishin, A.Yu. Yeremin, Variable Block CG Algorithms for solving large sparse symmetric positive definite linear systems on parallel computers, I: General Iterative Scheme, SIAM J. Matrix Anal. Appl. 16 (1995) 1135–1153.
[10] M. Robbé, M. Sadkane, A convergence analysis of GMRES and FOM methods for Sylvester equations, Numer. Algorithms 30 (2002) 71–89.
[11] M. Robbé, M. Sadkane, Exact and inexact breakdowns in block versions of FOM and GMRES methods, see http://maths2.univ-brest.fr/recherche/prepub/2005/breakdowns.pdf.
[12] Y. Saad, Iterative methods for sparse linear systems, PWS Publishing, Boston, 1996.
[13] V. Simoncini, E. Gallopoulos, Convergence properties of block GMRES and matrix polynomials, Linear Algebra Appl. 247 (1996) 97–119.
[14] B. Vital, Étude de quelques méthodes de résolution de problèmes linéaires de grande taille sur multiprocesseur, Ph.D. dissertation, IRISA, France, 1990.