Relaxation strategies for nested Krylov methods

Jasper van den Eshof\textsuperscript{a}, Gerard L.G. Sleijpen\textsuperscript{b,}\textsuperscript{*}, Martin B. van Gijzen\textsuperscript{c}

\textsuperscript{a}Department of Mathematics, University of Düsseldorf, Universitätsstr. 1, D-40224, Düsseldorf, Germany
\textsuperscript{b}Department of Mathematics, Utrecht University, P.O. Box 80.010, NL-3508 TA Utrecht, The Netherlands
\textsuperscript{c}CERFACS, 42 Avenue Gaspard Coriolis, 31057 Toulouse, Cedex 01, France

Received 14 February 2004; received in revised form 16 June 2004

Abstract

This paper studies computational aspects of Krylov methods for solving linear systems where the matrix–vector products dominate the cost of the solution process because they have to be computed via an expensive approximation procedure. In recent years, so-called relaxation strategies for tuning the precision of the matrix–vector multiplications in Krylov methods have proved to be effective for a range of problems. In this paper, we will argue that the gain obtained from such strategies is often limited. Another important strategy for reducing the work in the matrix–vector products is preconditioning the Krylov method by another iterative Krylov method. Flexible Krylov methods are Krylov methods designed for this situation. We combine these two approaches for reducing the work in the matrix–vector products. Specifically, we present strategies for choosing the precision of the matrix–vector products in several flexible Krylov methods as well as for choosing the accuracy of the variable preconditioner such that the overall method is as efficient as possible. We will illustrate this computational scheme with a Schur-complement system that arises in the modeling of global ocean circulation.

© 2004 Elsevier B.V. All rights reserved.

Keywords: Inexact matrix–vector multiplication; Flexible Krylov methods; Inner-outer iteration; Relaxation strategies; Schur complement

\textsuperscript{*}Corresponding author.
E-mail addresses: eshof@am.uni-duesseldorf.de (J. van den Eshof), sleijpen@math.uu.nl (G.L.G. Sleijpen), gijzen@cerfacs.fr (M.B. van Gijzen).

0377-0427/$ - see front matter © 2004 Elsevier B.V. All rights reserved.
1. Introduction

In this paper, we study iterative *Krylov subspace methods* for solving linear systems of the form

\[ Ax = b. \] (1.1)

More precisely, we will focus on problems for which the matrix–vector product is time-consuming and has to be computed via an approximation procedure. Examples of such type of problems include simulations in quantum chromodynamics [17], electromagnetic applications [5,13] and the solution of Schur-complement systems [3,16,24]. In these applications, the more accurately the matrix–vector product is approximated, the more expensive or time consuming the overall process becomes. The importance of these applications in scientific computing has resulted in various practical and theoretical studies on the impact of errors in the matrix–vector multiplies in Krylov subspace solvers, see e.g., [2,3,16,18]. Based on these investigations, so-called *relaxation strategies* have been proposed that aim to allow the error in the products to be as large as possible without compromising the accuracy of the method or its convergence speed too much.

In this paper, we focus on practical issues of dealing with expensive matrix–vector products. Assuming a simple theoretical cost model for the matrix–vector multiplications, we will argue that the gain of a relaxation strategies is limited. However, we will show that a significant further reduction in computation time can be achieved if the Krylov subspace solver is preconditioned with another Krylov subspace method (the preconditioning iteration), which is terminated at a much higher tolerance than the tolerance that is required in the final solution. The accurate, and hence expensive, matrix–vector products, which allow us to solve the system up to the desired tolerance, are performed in the outer iterations. The number of outer iterations can be decreased by increasing the efficiency of the preconditioner, but this comes at the cost of more expensive linear solves at each step. This leads to the central question of how accurately the linear systems of the preconditioning iteration need to be solved such that the overall work in the matrix–vector multiplications is as small as possible. We will study this questions using our cost model for the matrix–vector products, and we will argue that this nested approach can significantly reduces the overall amount of work in the matrix–vector products.

The Krylov methods that can be used for the outer iteration method are the so-called *flexible* Krylov methods, which allow for a variable preconditioner. We will discuss several possible methods and address the question of how accurately the matrix–vector products need to be computed in the outer iteration. Our theoretical work is confirmed by numerical experiments with the iterative solution of a Schur-complement system, for which each matrix–vector product requires the solution of a linear system.

The structure of this paper is as follows. After reviewing in Section 2 the main effects of inexact matrix–vector products on Krylov methods, we discuss the expected computational gain of using a relaxation strategy. We will argue that this is often modest in practice. This leads us to investigate a nested approach in which the outer Krylov subspace method is preconditioned by an inner Krylov subspace method that solves the preconditioning systems to relatively low accuracy, using less expensive matrix–vector products. In Section 3 we will explain the advantages of this approach and explain our ideas with Richardson iteration as outer iteration method. More advanced Krylov method for the outer iteration are analyzed in Section 4. In Section 5 we discuss related methods and ideas that can be found in literature. Section 6 gives numerical experiments with a Schur complement problem that stems from an ocean circulation model for steady barotropic flow, as described in [22].
2. Relaxation strategies and their computational gain

We want to find a vector, say \( \mathbf{x}' \), that approximately satisfies the Eq. (1.1) such that

\[
\| \mathbf{b} - \mathbf{A}\mathbf{x}' \|_2 = O(\varepsilon). \tag{2.1}
\]

Without loss of generality we assume that the vector \( \mathbf{b} \) is of unit length. The constant \( \varepsilon \) depends on the particular application and we assume it to be given. Instead of exact matrix–vector multiplications \( \mathbf{A}\mathbf{v} \) we will consider matrix–vector products that are calculated with some procedure that computes an approximation \( \mathcal{A}_\eta(\mathbf{v}) \) with precision \( \eta \) such that

\[
\mathcal{A}_\eta(\mathbf{v}) = \mathbf{A}\mathbf{v} + \mathbf{g} \quad \text{with} \quad \| \mathbf{g} \|_2 \leq \eta \| \mathbf{A} \|_2 \| \mathbf{v} \|_2. \tag{2.2}
\]

With an inexact Krylov subspace method we refer to the modified form of the method in which the exact matrix–vector products are replaced by approximate ones. Better approximations to the matrix–vector products (smaller \( \eta \)) can generally be achieved at higher cost. The natural question is how to choose the tolerances \( \eta \) in each step of the iteration method. A first idea to meet requirement (2.1) is to compute all matrix–vector products with an accuracy \( \eta = \varepsilon \). This can be seen as raising the unit roundoff to \( \varepsilon \) and this explains why, in most applications and for many Krylov methods, we ultimately obtain iterates that satisfy (2.1). Bouras and Frayssé reported various numerical results for GMRES in [2] with a precision for the matrix–vector product in step \( j + 1 \) that was essentially given by

\[
\eta_j = \frac{\varepsilon}{\| \mathbf{r}_j \|_2}. \tag{2.3}
\]

The vector \( \mathbf{r}_j \) is the residual vector computed in step \( j \) of the GMRES method. An interesting property of this empirical choice for \( \eta_j \) is that it requires very accurate matrix–vector products in the beginning of the process, while the precision is relaxed as soon as the residuals become increasingly smaller. This justifies their term relaxation strategy. For an impressive set of numerical experiments, they observe that the GMRES method with tolerance (2.3) converges about as fast as the unperturbed version, despite the, sometimes large, perturbations. Furthermore, the norm of the true residual (\( \| \mathbf{b} - \mathbf{A}\mathbf{x}_j \|_2 \)) seems to stagnate around a value of \( O(\varepsilon) \). Two recent publications, [18] and [16], have given theoretical explanations for the practical success of this approach and have shown that this strategy is in essence also appropriate for a much larger class of Krylov methods.

We now discuss the question of how much can be gained by applying such a relaxation strategy. In general, this is a difficult question, since often it is not known in advance how much time a matrix–vector multiplication takes. This does not only depend on the particular application but also on the vector in the multiplication, and on the required precision. For the sake of argument we therefore assume for the moment that the time \( T_{\text{mv}} \) spent in a matrix–vector multiplication as in 2.2 with an accuracy of \( \eta \) is given by

\[
T_{\text{mv}} = -c \log(\eta). \tag{2.4}
\]

This is a reasonable assumption if, for example, the matrix–vector product is approximated with an iterative solver that converges linearly. Since we are interested in the relative improvements that relaxation strategies can bring, the constant \( c \) is not of importance and is taken one in (2.4). It is also reasonable to assume that in every step the cost of the matrix–vector product dominates the other costs.
If we assume that the convergence behavior of the method is not influenced by the choice of the particular strategy used for the tolerances of the matrix–vector products, we have that the costs for \( k \) steps of an inexact Krylov method like inexact GMRES with a fixed tolerance \( \eta_j = \varepsilon \), and with a relaxed tolerance as in (2.3) are respectively given by

\[
C_f = - \sum_{j=0}^{k-1} \log(\varepsilon), \quad C_r = - \sum_{j=0}^{k-1} (\log(\varepsilon) - \log(\|r_j\|_2)).
\]  

(2.5)

It is standard practice to visualize the convergence history of iterative solvers by making a log-plot of the norms of the residuals versus the iteration number. This shows that both formulas have a visually appealing interpretation: Eq. (2.5) shows that the cost of the relaxation strategy approximates the area between the convergence curve and the constant line \( \varepsilon \) whereas for the fixed strategy the cost is (approximately) the size of the area between the lines \( \|r_0\|_2 \) and \( \varepsilon \). We give a simple illustration in Fig. 1 for a matrix from the matrix market [1].

In order to get some further insight into the expected computational gain of using a relaxation strategy compared to using a fixed precision for the matrix–vector products, let us assume that the convergence speed of the Krylov subspace method is described by the following simple expression \( \|r_j\|_2 = \varepsilon^{1/\beta} \) for some \( 0 < \alpha < 1 \) and \( \beta > 0 \). This expression captures the typical behavior of several types of convergence. For instance, with \( \beta = 1 \) the expression models linear convergence, whereas for \( \beta > 1 \) the convergence is superlinear. With this assumption the number of required iterations is approximately \( \kappa = (\log(\varepsilon)/\log(\alpha))^{1/\beta} \) and we get the following estimate for the ratio of the cost of both strategies, assuming that there is no change in convergence behavior,

\[
\frac{C_r}{C_f} \approx \int_0^\kappa \frac{\log(\varepsilon/\alpha^x)}{\log(\varepsilon)} \, dx / \int_0^\kappa \log(\varepsilon) \, dx = \frac{\beta}{1+\beta}.
\]

This indicates that if convergence is linear (\( \beta = 1 \)) then the improvement will be about a factor two. For many practical applications the convergence of GMRES is superlinear, which means that the norm of the residual decreases much faster in the final iterations than in the initial ones (i.e., \( \beta > 1 \)). Although superlinear convergence is a fortunate event, the advantage of relaxing the accuracy of the matrix–vector
product becomes less. This indicates that in many practical applications using a relaxation strategy leads to a limited improvement. We stress that this observation is not a particular property of the GMRES method with approximate matrix–vector multiplies but is also witnessed for other inexact Krylov methods.

3. Nested inexact Krylov subspace method

Stimulated by the observations in the previous section, we investigate how the computational work on the matrix–vector products can be reduced by preconditioning the inexact Krylov subspace methods by an inexact Krylov subspace method. For this purpose, we need the following notation: when an inexact Krylov subspace method is used for solving a linear system of the form $Az = y$ with a relative residual precision of at least $\varepsilon$, we will write

$$z = \mathcal{P}_\varepsilon(y), \quad \text{where } z \text{ satisfies } \|y - Az\|_2 \leq \varepsilon \|y\|_2.$$  

This operation can be used as a sort of flexible preconditioner that may change every step, depending not only on $\varepsilon$, but also on $y$. We will frequently call the inexact Krylov method and its variable preconditioner the outer iteration and preconditioning iteration, respectively. We will refer to the overall approach as a nested Krylov method in order to indicate the difference with the standard relaxed methods reviewed in the previous section.

In the outer iteration of our nested scheme, accurate matrix–vector products are required in order to achieve the target tolerance $\varepsilon$. The precision of the matrix–vector products in the preconditioning iterations can be modest, and depends on the accuracy $\varepsilon$ of the preconditioning iterations. Therefore, a trade-off has to be made between the accuracy and cost of the preconditioner and the number of outer iterations. In this section we discuss the use of the Richardson iteration method as the outer iteration method. In the next subsection we will discuss how accurately the matrix–vector products should be computed in the outer iteration. The trade-off that has to be made in choosing $\varepsilon_j$ is discussed in more detail in Section 3.2.

3.1. The outer iteration: Richardson iteration

The nested inexact Krylov subspace method with Richardson iteration as the outer iteration method is, for $j = 1, 2, \ldots, k$, defined by the following recurrences

$$z_{j-1} = \mathcal{P}_{\varepsilon_{j-1}}(r_{j-1}),$$
$$x_j = x_{j-1} + z_{j-1},$$
$$r_j = r_{j-1} - \mathcal{A}_{\eta_{j-1}}(z_{j-1}),$$

(3.1)

starting with $x_0 = 0$ and $r_0 = b$. The first line is the preconditioning operation where the linear system is solved with an inexact Krylov subspace method with a relative residual precision of $\varepsilon_{j-1}$. In the third line we have used the notation introduced in (2.2) to indicate that the matrix–vector product is approximately computed with a precision of $\eta_{j-1}$.

Due to the errors of the matrix–vector products, the ‘residual’ vector $r_k$ and the true residual $b - Ax_k$ drift apart during the iteration process. A straightforward estimate on the size of the true residual is given by

$$\|b - Ax_k\|_2 \leq \|r_k\|_2 + \|r_k - (b - Ax_k)\|_2.$$  

(3.2)
The second quantity on the right is commonly referred to as the norm of the residual gap. An inductive argument using the recursions in (3.1) shows that
\[
\|r_k - (b - Ax_k)\|_2 \leq \|r_{k-1} - (b - Ax_{k-1})\|_2 + \eta_{k-1}\|A\|_2\|z_{k-1}\|_2 \\
\leq \|A\|_2 \sum_{j=0}^{k-1} \eta_j \|z_j\|_2.
\] (3.3)

For the computed residuals we easily find that
\[
\|r_k\|_2 \leq \xi_{k-1}\|r_{k-1}\|_2 + \eta_{k-1}\|z_{k-1}\|_2\|A\|_2 \\
\leq \prod_{i=0}^{k-1} \xi_i + \|A\|_2 \sum_{j=0}^{k-1} \eta_j \|z_j\|_2 \prod_{i=j+1}^{k-1} \xi_i.
\] (3.4)

Since the products of the \(\xi_i\) in (3.4) can become arbitrarily small by increasing \(k\), this relation can be used to show convergence to zero of the computed residuals. Eq. (3.3), on the other hand, shows that the size of the residual gap determines the stagnation level of the method. This is a typical and fundamental observation on which the analysis for a large class of Krylov subspace methods in [18] is based. Following that paper we propose to take \(\eta_j\) such that the norm of the residual gap does not become larger than the order of \(\varepsilon\). In order to obtain a practical strategy we use the estimate
\[
\|z_j\|_2 \leq (1 + \xi_j)\|A^{-1}\|_2\|r_j\|_2 \leq 2\|A^{-1}\|_2\|r_j\|_2.
\] (3.5)

All together this suggests to choose \(\eta_j = \varepsilon / \|r_j\|_2\) as in (2.3). In this case (3.3) becomes
\[
\|r_k - (b - Ax_k)\|_2 \leq \varepsilon 2\|A\|_2\|A^{-1}\|_2.
\] (3.6)

We remark that one should be aware that the bound (3.6) depends on the inverse of the smallest singular value of the matrix \(A\) and not on the norm of the computed solution: there can be circumstances where a relaxation strategy based on \(z_j\) and an estimate for the norm of the solution is more desirable.

3.2. The choice of the precisions \(\xi_j\)

As mentioned in the introduction, the efficiency of the nested Krylov methods discussed in this paper depends on how accurately the preconditioning iteration solves the linear systems. A more accurate preconditioning iteration requires more accurate and hence more expensive matrix–vector products but also reduces the number of outer iterations in which accurate matrix–vector products are necessary. On the other hand, a less accurate preconditioning iteration can be accomplished with less accurate matrix–vector products in the preconditioning iterations, but at the cost of more outer iterations. Therefore, we want to provide some insight into the problem of selecting the tolerances \(\xi_j\). Making general statements about the optimal sequence of these tolerances is difficult since this requires detailed knowledge of the convergence behavior of Krylov subspace methods and of the cost of the matrix–vector multiplications. We assume that the time for a matrix–vector multiplication with a precision \(\eta\) is given by (2.4), where for simplicity we again take the constant \(c\) to be one. Furthermore, the required residual precision is set to \(\varepsilon = 10^{-l}\) and we assume that the residual reduction in step \(j\) of the outer iteration is precisely equal to \(\xi_{j-1}\).
This means that the norm of the residual at the end of step $j$ is equal to $\varepsilon_j$ with

$$1 = \varepsilon_0 \geq \varepsilon_1 \geq \ldots \geq \varepsilon_k = \varepsilon = 10^{-l},$$

and $\varepsilon_j = \prod_{i=0}^{j-1} \xi_i$.

We use the Richardson iteration as the outer iteration. If we assume that the variable preconditioner is a linearly converging method we can give a visual illustration for the cost of the preconditioning iterations as in Section 2. Fig. 2 shows this for the case we use a relaxation strategy in the preconditioning iterations and $\xi_0 = 10^{-2}, \xi_1 = 10^{-1}, \xi_2 = 10^{-2}, \xi_3 = 10^{-3}$.

If we write $\varepsilon_j = 10^{-t_j}$ for appropriate $t_j$ (hence $t_0 = 0$ and $t_k = l$) and assume that the total number of matrix–vector products in the preconditioning iterations is equal to $m$, then we have the following estimate for the total cost of the matrix–vector multiplications:

$$\tilde{C}_k = \frac{m}{2l} \sum_{i=1}^{k} (t_i - t_{i-1})^2 + \sum_{i=0}^{k-1} (l - t_i). \tag{3.7}$$

The first term represents the cost of the preconditioning iterations and the second term represents the cost of the matrix–vector products in the outer iteration. The optimal values of $t_i$ are given by the following lemma.

**Lemma 3.1.** For fixed $k < \frac{1}{2} (1 + \sqrt{1 + 8m})$ the quantity $\tilde{C}_k$ in (3.7) is minimized by

$$t_i = \frac{l}{2m} i(k - i) + \frac{l}{k} i \quad \text{for } i \in \{0, \ldots, k\}. \tag{3.8}$$

Furthermore, the optimal value of $k$ is given by the largest integer strictly smaller than $\frac{1}{2} (1 + \sqrt{1 + 8m})$. 

Fig. 2. The gray area approximates the cost of the preconditioning iterations when convergence is linear. The horizontal axis gives the cumulative number of inner iterations, and the vertical axis the 10-log of the norm of the residual in the inner iteration.
Proof. Differentiating (3.7) with respect to $t_i$ (for $i \in \{1, \ldots, k-1\}$) and equating to zero gives the equation

$$
\frac{m}{l} (-t_{i-1} + 2t_i - t_{i+1}) - 1 = 0.
$$

Using standard theory for solving recurrences we get that $t_i = -\frac{l}{2m} i^2 + \alpha i + \beta$ for some constants $\alpha$ and $\beta$. Using the boundary conditions $t_0 = 0$ and $t_k = l$ we find expression (3.8). It can be easily verified that if $k < \frac{1}{2}(1 + \sqrt{1 + 8m})$ then all the $t_i$ in (3.8) are in the open interval $(0, l)$ and therefore are the optimal points. Furthermore, we have that the minimal value of $\tilde{C}_k$ is smaller than or equal to the minimal value of $\tilde{C}_{k-1}$ (to see this, select for $\tilde{C}_k$ the $t_j$ equal to the optimal points for $\tilde{C}_{k-1}$ and choose the additional point equal to $l$). Suppose that $k'$ is the largest integer strictly smaller than $\frac{1}{2}(1 + \sqrt{1 + 8m})$ then at least one of the global minimizers of $\tilde{C}_{k'+1}$ defined by (3.8) is outside the interval $(0, l)$, and therefore the minimum is attained at the boundary of the interval (that is $t_i = l$ for one or more indices $i$). The minimal value of $\tilde{C}_{k'}$ equals the minimal value of $\tilde{C}_{k'+1}$. The optimal number of (nonzero) tolerances is given by $k = k'$. □

This lemma implies that the preconditioner should be chosen most accurately in the early iterations which can be explained by the fact that at this point the matrix–vector products in the outer iteration are most expensive. It is interesting to notice that in case the total number of preconditioning iterations, $m$, is large compared to the number of outer iterations $k$, we have for the optimal $t_i$ that $t_i \approx il/k$. This implies that keeping $\xi_{j-1}$ constant is almost optimal. For this reason we will use $\xi_j = \xi$ in our numerical experiments in Section 6.

The discussion in this section gives simple, but visually appealing, insight into the computational advantage of the nested preconditioning paradigm. We should remark that in practice the assumption that the solver that acts as preconditioner is converging linearly is overly simplistic. A drawback of the variable preconditioning is, compared to the standard relaxed method discussed in Section 2, that it increases the total number of matrix–vector products (although it reduces the number of accurate matrix–vector products). On the other hand, an advantage is that for the standard relaxed methods the upper bounds on the residual gap, for example the ones derived in [18], show a dependence on the number of iterations. This means that there can be a serious accumulation of errors in the residual gap if the number of iterations is large, which in turn may require more accurate matrix–vector products, see also [16]. We note that in the nested scheme the preconditioning iterations (greatly) reduce the required number of outer iterations and consequently also the accumulation of errors in the residual gap.

4. Optimal Krylov methods

In the previous section we discussed improving the efficiency of relaxation strategies for Krylov subspace methods by using the relaxed Krylov method as a preconditioner for Richardson iteration. In this section we discuss the obvious generalization of this scheme to more advanced Krylov methods for the outer iteration. Methods that can be used here should be able to handle variable preconditioning. Therefore, we restrict our attention to Flexible GMRES and GMRESR.
4.1. Flexible GMRES as outer iteration method

The Flexible GMRES method (FGMRES) by Saad [15] is a variant of the GMRES method that can deal with variable preconditioning. In this method an orthogonal basis \( v_0, \ldots, v_{k-1} \) for the span of \( v_0, Az_0, \ldots, Az_{k-1} \) is constructed where \( z_j = P_{\xi_j}(v_j) \). Defining the matrices \( V_k \) and \( Z_k \) with as \( j+1 \)-st column, respectively, \( v_j \) and \( z_j \), we can summarize this process with inexact matrix–vector products by the relations

\[
AZ_k + F_k = V_{k+1}T_k, \quad x_k = Z_kT_k^+e_1 \quad \text{and} \quad r_k = V_{k+1}(I - T_kT_k^+)e_1, \tag{4.1}
\]

where \( T_k^+ \) is the Moore–Penrose inverse of \( T_k \). For more details on these relations (for exact matrix–vector product) we refer to [15, Section 2.2]. Note that in (4.1) we have an additional perturbation \( F_k \) which originates from the fact that in every step of FGMRES we make an error in the matrix–vector multiplication with the vectors \( z_j \). We therefore have that

\[
\|F_k T_j^{++}e_1\|_2 \leq \eta_j \|A\|_2 \|z_j\|_2.
\]

To derive a strategy for the tolerances for the errors in the matrix–vector multiplications in FGMRES we will follow the approach taken in [18] and Section 3.1 for Richardson iteration. This means that we choose the tolerances in such a way that the norm of the residual gap does not exceed the order \( \varepsilon \). Using (4.1) we find that

\[
\|r_k - (b - Ax_k)\|_2 = \|F_k T_k^{++}e_1\|_2 \leq 2\|A\|_2 \|A^{-1}\|_2 \|T_k^{++}\|_2 \sum_{j=0}^{k-1} \eta_j \|r_j\|_2.
\]

For the estimate we expressed \( F_k T_k^{++}e_1 \) as \( \sum_{j<k} F_k e_{j+1} e_{j+1} T_k^{++} e_1 \) and we used the bound (3.5). As for the standard inexact GMRES method analyzed without variable preconditioning in [18,16], the norm of \( T_k^{++} \) is difficult to bound a priori. In the exact case (i.e., \( \eta_j = 0 \) for all \( j \)), we have for small enough precisions \( \xi_j \):

\[
\|T_k^{++}\|_2^{-1} = \sigma_{\text{min}}(T_k) = \sigma_{\text{min}}(V_k + (AZ_k - V_k)) \geq 1 - \sqrt{k} \max_j \xi_j,
\]

with \( \sigma_{\text{min}}(\cdot) \) denoting the smallest singular value of a matrix. For this reason we assume that \( \|T_k^{++}\|_2 \) is bounded by a modest constant in the remainder of this section. With this assumption we have that the relaxation strategy for inexact GMRES given by (2.3) is also useful in the flexible context and leads to

\[
\|r_k - (b - Ax_k)\|_2 \leq 2k\varepsilon \|A\|_2 \|A^{-1}\|_2 \|T_k^{++}\|_2.
\]

These arguments show that we do not expect accuracy problems when FGMRES is used in the outer iteration.

A notable difference with Richardson iteration described in the previous section is that in FGMRES the preconditioner is applied to multiples of the residuals of the corresponding Galerkin approximations (also referred to as the approximations from the associated flexible FOM process). For this reason the residual reduction of FGMRES can be less than \( \xi_j \) in step \( j+1 \) if the residual of the corresponding Galerkin approximation is large at this point. This observation was made earlier by Vuik [23] and is not difficult to understand by assuming exact matrix–vector products: if we define \( \varepsilon \equiv e_{j+1}^+(I - T_j T_j^{-1})e_1 \),
then \(|z|\) is the norm of the Galerkin residual, and, using the optimality of FGMRES, we find with \(y_{j+1} \equiv [(T_j^{-1}e_1)^T, z]^T\) the sharp estimate

\[
\|r_{j+1}\|_2 \leq \|b - A Z_{j+1} y_{j+1}\|_2 = \|b - A Z_j T_j^{-1} e_1 - A z_j + 1\|_2 \\
= \|v_{j+1} - Az_{j+1}\|_2 \leq \xi_j |z|.
\]

(4.2)

This type of results can be found in [23]. It shows that in case the associated FOM process suffers from near breakdowns, i.e., the convergence curve shows large peaks, then FGMRES might not be a good choice: the residual reduction in FGMRES can be much less than what can be expected from the reductions obtained in the preconditioning iterations. However, if we have that \(\xi_j = \xi\) for all \(j\) and \(\xi\) is small enough then this is not a serious problem as the following lemma demonstrates. For simplicity we have again assumed exact matrix–vector products. For ease of interpretation we mention that for Richardson iteration with exact matrix–vector products, we have that \(\|r_k\|_2 \leq \prod_{j=0}^{k-1} \xi_i\).

**Lemma 4.1.** Let \(\xi_j = \xi < \frac{1}{2}\) and \(\eta_j = 0\) for all \(j\). Then the residuals of the FGMRES method satisfy

\[
\|r_k\|_2^2 \leq \xi^{2k} \beta\left(\left(\frac{1 + \beta}{2}\right)^{k+1} - \left(\frac{1 - \beta}{2}\right)^{k+1}\right)^{-1}
\]

with \(\beta \equiv \sqrt{1 - 4\xi^2}\).

**Proof.** We have that

\[
\frac{1}{\|r_{k+1}\|_2^2} \geq \frac{1}{\xi^2} \frac{1}{\beta^2} = \frac{1}{\xi^2} \left(\frac{1}{\|r_k\|_2^2} - \frac{1}{\|r_{k-1}\|_2^2}\right),
\]

(4.3)

where for the inequality we have used 4.2 and for the equality we have used a well-known relation due to Brown [4, Section 5]. We define \(\gamma_k = \xi^{2k}/\|r_k\|_2^2\). From (4.3) it follows that \(\gamma_{k+1} \geq \gamma_k - \xi^2 \gamma_{k-1}\) and \(\gamma_0 = 1, \gamma_1 \geq 1\). Furthermore, we introduce the quantity \(\rho_k\) that satisfies the recursion \(\rho_{k+1} = \rho_k - \xi^2 \rho_{k-1}\) with \(\rho_0 = \rho_1 = 1\). Our first step is to show that for all \(k\) we have that \(0 \leq \rho_k \leq \gamma_k\). If \(\xi \leq \frac{1}{2}\) then there exist constants \(\mu, v \in [0, 1]\) such that \(\mu + v = 1\) and \(\mu v = \xi^2\). Hence,

\[
(\gamma_{k+1} - \mu \gamma_{k-1}) \geq v(\gamma_k - \mu \gamma_{k-1}), \quad \gamma_1 - \mu \gamma_0 = \rho_1 - \mu \rho_0 = 1 - \mu \geq 0,
\]

\[
(\rho_{k+1} - \mu \rho_{k-1}) = v(\rho_k - \mu \rho_{k-1}).
\]

This shows by induction that \((\gamma_k - \mu \gamma_{k-1}) \geq (\rho_k - \mu \rho_{k-1}) \geq 0\). Hence, \(\gamma_k \geq \mu \gamma_{k-1} + (\rho_k - \mu \rho_{k-1})\). Again with induction, we find that \(\gamma_k \geq \mu \rho_{k-1} + (\rho_k - \mu \rho_{k-1}) = \rho_k \geq 0\). The proof is concluded by solving the recursion for the \(\rho_k\) using standard techniques. \(\Box\)

This lemma demonstrates that \(\xi^{-k} \|r_k\|_2\) approaches a value smaller than or equal to one for \(\xi\) going to zero and, therefore, the disadvantage of FGMRES that the residual reduction can be much less than \(\xi\) is guaranteed not to be a problem in the context of nested Krylov schemes if we work with modest values of the \(\xi_j\).

### 4.2. The outer iteration: GMRESR

The GMRESR method of Van der Vorst and Vuik [21] is another variant of the GMRES method that allows for variable preconditioning. It was originally proposed with the GMRES method for the
preconditioning iteration as a way to improve the efficiency of the restarted GMRES method. In this paper we consider GMRESR with any Krylov method for the preconditioning iteration, nevertheless, we refer to this method as GMRESR. In the GMRESR method the flexible preconditioner is directly applied to the lastly computed residual so that we have \( z_j = P_{\xi_j}(r_j) \). The GMRESR method minimizes the residual by constructing its iterates as a suitable linear combination of all previously computed vectors \( z_j \). Therefore, a simple argument shows that this guarantees in the exact case a residual reduction of at least \( \xi_j \) in step \( j + 1 \) which is an advantage over the use of FGMRES as discussed in the previous section.

We now discuss the matrix formulation of the GMRESR method with inexact matrix–vector products. In the inexact GMRESR method in every step decompositions are updated such that after \( k \) steps of the method we have that

\[
AZ_k + F_k = C_k B_k, \quad Z_k = U_k B_k, \quad \text{with } C_k^* C_k = I_k \quad \text{and } B_k \text{ upper triangular.}
\]  

(4.4)

As in the previous section we denote matrices by capitals and their \( j \) + 1-st column by a small letter with subscript \( j \). Hence, \( U_k e_{j+1} = u_j \). In the second part of a GMRESR iteration step, the residual and iterate are updated as follows:

\[
x_k = x_{k-1} + u_{k-1}(c_k^* r_{k-1}), \quad r_k = r_{k-1} - c_{k-1}(c_k^* r_{k-1}).
\]  

(4.5)

Notice that, in contrast to a standard implementation of GMRESR, we assumed here that the vectors \( c_j \) are normalized which is not an essential restriction. The last two relations can be summarized by

\[
C_k D_k = R_{k+1} J_k \quad \text{and} \quad U_k D_k = X_{k+1} J_k,
\]

where \( D_k = \text{diag}(c_0^* r_0, \ldots, c_{k-1}^* r_{k-1}) \), \( J_k \) is the \((k+1) \times k\) matrix with ones on its main diagonal and minus ones on its subdiagonal. With \( J_k \) the upper \( k \times k \) block of \( J_k \), substitution and using that \( D_k J_k^{-1} e_1 = D_k \tilde{I} = C_k^* r_0 \) (since \( c_j^* r_j = c_j^* r_0 \)) yields

\[
AZ_k + F_k = R_{k+1}(J_k D_k^{-1} B_k) \quad \text{and} \quad x_k = U_k(C_k^* r_0) = Z_k(J_k D_k^{-1} B_k)^{-1} e_1.
\]

As in the previous sections, we can use these relations to determine an expression for the residual gap:

\[
\|r_k - (b - A x_k)\|_2 = \|F_k S_k^{-1} e_1\|_2.
\]

This shows that the sensitivity for errors in the matrix–vector multiplications (which result in the perturbation term \( F_k \)) is determined by the size of the elements of the vector \( S_k^{-1} e_1 \). For this vector we have that

\[
S_k^{-1} e_1 = B_k^{-1} D_k J_k^{-1} e_1 = B_k^{-1} C_k^* r_0 = (AZ_k + F_k)^* r_0.
\]

The size of the elements of this vector are difficult to assess. Intuitively it is not difficult to understand that small errors in the matrix–vector product can have a considerable influence on the residual gap and, therefore, on the stagnation level of the method. This happens if the vector that results from the perturbed matrix–vector product \( Az_k + f_k \) makes a small angle with the span of the previously computed vectors. Orthogonalizing this vector with respect to the previously computed vectors yields a vector that is for a relatively large part contaminated by the error in the matrix–vector product. Such an unfortunate cancellation is reflected by large elements in the vector \( S_k^{-1} e_1 \). We are interested in the norm of the residual gap and we will investigate the size of the elements of the vector \( S_k^{-1} e_1 \) in the remainder of this section.
In [19], Section 3.2, bounds are given for the elements of the vector $|S_k^{-1}e_1|$ for the GMRESR method without preconditioning, in which case GMRESR reduces to the GCR method. This paper shows that if the convergence curve of the associated FOM process exhibits large peaks, then the elements of the vector $|S_k^{-1}e_1|$ are very large. Numerical experiments in [19] confirm that in this case GMRESR is indeed extremely sensitive for errors in the matrix–vector multiplies.

In our computational scheme, GMRESR is used in the outer iteration with variable preconditioning. Here we work with a preconditioner that reduces the residual in step $j < j$ for $i < j$. We have in step $j + 1$ that $Az_j = r_j + (Az_j - r_j)$ with $\|Az_j - r_j\|_2 \leq \xi \|r_j\|_2$. Since, $C_j^*r_j = \tilde{0}$ we find for the diagonal elements $|B_k|_{jj} \geq \|r_j-1\|_2(1-\xi)$ and for the off-diagonal elements $|B_k|_{ij} \leq \|r_j-1\|_2\xi$ for $i < j$. We can use this to show that

$$|B_k^{-1}| \leq \frac{1}{1-\xi} \text{diag}(\|r_0\|_2, \ldots, \|r_{k-1}\|_2)^{-1}\tilde{B}_k^{-1},$$

where the matrix $\tilde{B}_k$ is upper triangular with ones on its main diagonal and $-\alpha$ in all its off-diagonal entries with $\alpha \equiv \xi/(1-\xi)$. It can be shown that $(\tilde{B}_k^{-1})_{ij}$ equals one for $i = j$ and equals $\alpha(1+\alpha)^{j-i}$ for $i < j$ (see also [12, Eq. (8.3)]). The vector $e_j^*\tilde{B}_k^{-1}$ is zero in the first $j$ components and we can show that

$$\|\tilde{B}_k^{-*}e_{j+1}\|_1 = 1 + \alpha \sum_{i=0}^{k-j-2} (1+\alpha)^i = (1+\alpha)^{k-j-1} = (1-\xi)^{j+1-k}.$$

The proof is completed as follows:

$$|e_j^*S_k^{-1}e_1| \leq |e_j^*B_k^{-1}| |C_k^*r_0| \leq (1-\xi)^{j-k} \|r_j\|_2^{-1} \max_{i>j} |c_i^*r_0| \leq (1-\xi)^{j-k}. \quad \Box$$

This result shows that we can safely use the relaxation strategy (2.3) for inexact GMRESR when the preconditioner leads to a residual reduction in every step that is substantial enough.

Interestingly, we see that the observations for the GMRESR method are the converse of our findings for FGMRES in the previous section. There, we argued that a near breakdown of the associated FOM process results in a residual reduction that can be less than what can be expected from the reductions obtained in the preconditioning iteration, whereas accuracy problems are not to be anticipated. Conversely, in GMRESR problems can occur with the accuracy of the method in case of a near breakdown, while the residual reduction will be as achieved in the preconditioning iteration. However, for $\xi$ small enough in every step, we argued that the wavering in the residual reduction in FGMRES and the impact on the accuracy in GMRESR are insignificant. We conclude that the differences between both methods are small for $\xi$ small enough. The numerical experiments in Section 6 underline this conclusion.
5. Comparison with other approaches

Some inexact methods proposed in literature have important connections with the nested inexact Krylov subspace method of this paper. In this section we discuss some related methods and ideas.

Golub and his co-workers [10] introduced the successive Lanczos method for the computation of eigenvectors with the Lanczos method. For this method, the user has to specify in advance the tolerances $\varepsilon_j$ and the precise number of iterations for each inner iteration (there appears to be no automatic stopping condition for the inner iterations). At the beginning of each cycle the computed approximation from the previous cycle is used as a starting vector. In the presented numerical experiments the method was not combined with a relaxation strategy for the inner iterations (the possibility is mentioned).

The nested approach of this paper can be viewed as an improved version of related schemes in [11] and [9, Section 9] used in simulations in quantum chromodynamics with Neuberger fermions [14]. The forthcoming paper [6] presents numerical results for the QCD overlap operator using the nested scheme of this paper. Giusti et al. recently proposed [9, Section 9], what they call, an adapted-precision inversion algorithm for problems from quantum chromodynamics which they interpret as some form of iterative refinement. This is essentially equal to the method proposed here with Richardson iteration with directly computed residuals in the outer iterations. The authors do not discuss specific choices for the precision of the matrix–vector product in the outer iteration and use a fixed precision in the inner iteration.

Carpentieri [5] describes experiments with nested Krylov subspace methods that use inexact matrix–vector products. He uses flexible GMRES in the outer iteration and GMRES in the inner iteration. The matrix–vector products, which are computed using a fast multipole technique, are evaluated to a high precision in the outer iteration, whereas the matrix–vector products in the inner iteration are evaluated to a lower, but fixed, precision. In the numerical experiments that are described in the thesis no relaxation strategies are applied.

The use of inner iterations set to variable precisions in the context of flexible or variable preconditioning was investigated by Dekker [8] for a problem from domain decomposition.

6. Numerical experiments

6.1. An example from global ocean circulation

In this section we present our, preliminary, experiences with nested inexact Krylov subspace methods for solving Schur complement systems. The example comes from a finite element discretization of a model that describes the steady barotropic flow in a homogeneous ocean with constant depth and in nearly equilibrium as described in [22]. The model is described by the following set of partial differential equations:

$$
-r \nabla^2 \psi - \beta \frac{\partial \psi}{\partial x} - \kappa \nabla^2 \zeta = \nabla \times F \quad \text{in } \Omega
$$

$$
\nabla^2 \psi + \zeta = 0,
$$

in which $\psi$ is the stream-function and $\zeta$ the vorticity. The domain $\Omega$ is the part of the world that is covered by sea. The external force field $F$ is proportional to the wind stress. The other parameters in these equations
are: the lateral viscosity $\alpha$, the bottom friction $r$ and the Coriolis parameter $\beta$. The above equations are complemented by a suitable set of (no-slip) boundary conditions.

Discretization with the method described in [22] leads to the following linear system of equations

$$
\begin{bmatrix}
 rL - C & \alpha\tilde{L} \\
 -\tilde{L}^* & M
\end{bmatrix}
\begin{bmatrix}
 \psi \\
 \zeta
\end{bmatrix} =
\begin{bmatrix}
 f \\
 0
\end{bmatrix}.
$$

(6.1)

In this expression the discrete counterparts of the continuous operators can be recognized, these are

$$
-\nabla^2 \rightarrow L, \tilde{L}, \tilde{L}^*, \quad \beta \frac{\partial}{\partial x} \rightarrow C, \quad 1 \rightarrow M.
$$

The matrices $L, \tilde{L},$ and $\tilde{L}^*$ differ due to the incorporation of the boundary conditions. The mass matrix $M$ is lumped and, therefore, a diagonal matrix.

The physical parameters are chosen as in Section 7.1 in [22] except for the viscosity parameter $\alpha$ which we take $10^5$ in our experiments. A contour plot of the stream-function $\psi$ for these parameters is given in Fig. 3. The resolution is set to two degrees which results in a matrix of dimension 26455.

6.2. The Schur complement systems

From Eq. (6.1) we can eliminate either $\psi$, which gives the Schur complement for $\zeta$:

$$
(M + \alpha \tilde{L}^* (rL - C)^{-1} \tilde{L}) \zeta = \tilde{L}^* (rL - C)^{-1} f,
$$

or we can eliminate $\zeta$, which gives the Schur complement for $\psi$:

$$
((rL - C) + \alpha \tilde{L} M^{-1} \tilde{L}^*) \psi = f.
$$

(6.3)

The equation for the stream-function has the obvious advantage that, since $M$ is diagonal and hence trivially invertible, operations with the Schur complement matrix $(rL - C) + \alpha \tilde{L} M^{-1} \tilde{L}^*$ are relatively
cheap. This in contrast to the equation for the vorticity (6.2), where operations with the Schur complement require the solution of a linear system with the matrix $rL - C$.

There are, however, reasons why it can be preferable to solve (6.2) instead of (6.3). The Schur complement for the stream-function can be considerably worse conditioned than the Schur complement for the vorticity, in particular if $\varepsilon$ is large, or if the mesh size $h$ is small. The ill-conditioning of the stream-function Schur complement is caused by the term $\tilde{L}M^{-1}\tilde{L}^*$, which is a discretized biharmonic operator, of which the condition number is $O(h^{-4})$. In practice it is very difficult to derive effective preconditioners for this operator. On the other hand, the diagonal matrix $M$ turns out to be a very effective preconditioner for (6.2) and $rL - C$ is a discretized convection–diffusion operator for which also reasonably effective preconditioners are readily available. The smaller number of iterations for solving (6.2) in combination with the existence of preconditioners for the convection–diffusion operator may outweigh the extra work for the more costly matrix–vector products. This will be illustrated with the numerical experiments that are described below.

Another advantage of solving for the vorticity is that, since $M$ is trivially invertible, we can construct solutions $\xi$ for various values of $\varepsilon$ by constructing the Krylov subspace only once using ideas from the so-called class of multi-shift Krylov subspace methods, e.g., [7] and [16, Section 10].

6.3. Numerical results for the vorticity Schur complement

If (6.2) is solved using a Krylov subspace method then a system with discrete convection–diffusion operator $rL - C$ has to be solved for every matrix–vector product with the Schur complement. In our experiments this was done using Bi-CGSTAB [20] with an incomplete LU preconditioner. The Bi-CGSTAB method was terminated when a relative residual precision of $\eta$ was achieved. Note that it follows from the analysis in [16, Section 8] that this does not guarantee a relative error of $\eta$ for the matrix–vector product and ideally $\eta$ should be taken somewhat smaller.

We aim for a residual precision of about $10^{-5}$ and consider a few different approaches for solving (6.2). In order to make a fair comparison between different methods, the parameter $\varepsilon$ was empirically chosen such that all the methods achieve about the same accuracy. For the first method we have applied the inexact (full) GMRES method with the relaxation strategy of Bouras and Fraysse (2.3). As a preconditioner we have used the diagonal matrix $M$. This preconditioner corrects for the adverse scaling effects introduced by the use of spherical coordinates and becomes increasingly more effective for smaller $\varepsilon$. The results for this strategy are plotted in Fig. 4. On the horizontal axis the total number of iterations with Bi-CGSTAB is given as a measure for the amount of work spent in the matrix–vector product. The vertical axis gives the norm of the true residual. The number of GMRES iterations is large for this problem, about 130, which deteriorates the precision of the inexact Krylov method because of the accumulation in the residual gap of the errors in the matrix–vector product. For this reason we have chosen the empirical value $\varepsilon = 10^{-7}$.

The convergence of GMRES is linear for the above example. The gain by applying the relaxation strategy is about a factor of two: the number of iterations Bi-CGSTAB drops from 2000 to about 900. This experimentally observed gain is consistent with the theoretical prediction for the gain that is presented in Section 2 (although the number of necessary Bi-CGSTAB iterations is not proportional to $-\log(\eta)$). As a consequence of the relaxation strategy, the number of Bi-CGSTAB iterations drops from 38.5 for the initial GMRES iterations to 0.5 for the last. Note that Bi-CGSTAB allows for two tests of the residual norm, and hence may terminate halfway an iteration. In the latter case this means that the required residual reduction was obtained after the first of the two matrix–vector multiplies in one iteration.
The alternative is to use the method from Section 4.1, that is, we precondition (inexact) FGMRES with an inexact GMRES method set to a precision of $\zeta = 10^{-1}$. For the GMRES methods in the outer and preconditioning iteration we have both used the Bouras–Frayssé relaxation strategy (2.3) with $\varepsilon$ respectively given by $10^{-5}$ and $10^{-1}$. The results are given in Fig. 4 where every ‘*’ corresponds to the true residual in one step of FGMRES in the outer iteration. For the nested method only a few outer iterations are necessary and therefore the residual gap is less contaminated by errors in the matrix–vector product which results in a final precision of almost $10^{-5}$. The large number of matrix–vector products and the accumulation of the errors now manifests itself in the preconditioning iterations: we do not achieve a residual reduction of $10^{-1}$ in every outer step. In this picture we have also included the results for a nesting strategy with GMRESR in the outer iteration, the norm of the true residuals is indicated with ‘o’.

The difference between the convergence of FGMRES and GMRESR is very small, which is in line with the remarks made at the end of the Sections 4.1 and 4.2.

To further illustrate the relaxation process in the outer iteration we have tabulated in Table 1 some interesting numerical values for the GMRESR-method. For the FGMRES method we find very similar

---

**Table 1**

<table>
<thead>
<tr>
<th>$j$</th>
<th>Tolerance ($\eta_{j-1}$)</th>
<th>Bi-CGSTAB iterations</th>
<th>$|b - Ax_j|_2$</th>
<th>$|r_j|_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$1.0 \cdot 10^{-5}$</td>
<td>35.5</td>
<td>$1.2 \cdot 10^{-1}$</td>
<td>$1.2 \cdot 10^{-1}$</td>
</tr>
<tr>
<td>2</td>
<td>$8.2 \cdot 10^{-5}$</td>
<td>33.5</td>
<td>$1.9 \cdot 10^{-2}$</td>
<td>$1.9 \cdot 10^{-2}$</td>
</tr>
<tr>
<td>3</td>
<td>$5.2 \cdot 10^{-4}$</td>
<td>28.5</td>
<td>$2.6 \cdot 10^{-3}$</td>
<td>$2.6 \cdot 10^{-3}$</td>
</tr>
<tr>
<td>4</td>
<td>$3.9 \cdot 10^{-3}$</td>
<td>7</td>
<td>$4.4 \cdot 10^{-4}$</td>
<td>$4.3 \cdot 10^{-4}$</td>
</tr>
<tr>
<td>5</td>
<td>$2.3 \cdot 10^{-2}$</td>
<td>3</td>
<td>$7.5 \cdot 10^{-5}$</td>
<td>$6.8 \cdot 10^{-5}$</td>
</tr>
<tr>
<td>6</td>
<td>$1.4 \cdot 10^{-1}$</td>
<td>1</td>
<td>$4.1 \cdot 10^{-5}$</td>
<td>$9.4 \cdot 10^{-6}$</td>
</tr>
</tbody>
</table>

---

Fig. 4. Norm true residual as function of the total number of iterations Bi-CGSTAB for the inexact GMRES method with fixed precision $\varepsilon = 10^{-6}$ (+), relaxed GMRES method with $\varepsilon = 10^{-7}$ (squares) and relaxed FGMRES preconditioned with relaxed GMRES set to a precision 0.1 (*) and the same for GMRESR in outer iteration (o).
numbers, which we have not reported here. The table shows for step $j$ the used (approximate) tolerance for the matrix–vector product $\eta_{j-1}$, the number of Bi-CGSTAB iterations to compute the matrix–vector product in step $j$ of the outer iteration and the norm of the true and computed residual at the end of step $j$ (recall that $b$ is normalized). With discretization step sizes that are more relevant in practice, the norm of the true residual will not be known during the process. The results in this table clearly illustrate that the norm of the true residual stagnates, in contrast to the norm of the updated residual. Another noteworthy observation is the sharp decrease in the number of Bi-CGSTAB iterations if the required tolerance is relaxed from $5.2 \cdot 10^{-4}$ to $3.9 \cdot 10^{-3}$. This is explained by the typical convergence behavior for Bi-CGSTAB that we observed for this example, which exhibits a fast decrease of the residual norm during the first iterations followed by a phase of slow convergence. The transition between fast and slow convergence is typically when the norm of the scaled residual is $\mathcal{O}(10^{-3})$.

A direct consequence of this initial fast convergence behavior of Bi-CGSTAB is that half a Bi-CGSTAB iteration (this is one application of the ILU-preconditioner and one matrix–vector product) is sufficient to reduce the scaled residual norm to below 0.1, which is an upper bound on the criterion for Bi-CGSTAB in the inner-loop. As a result, there is no practical difference between using a relaxation strategy or a fixed precision for the inner-iteration in this example.

6.4. Numerical results for the stream-function Schur complement

Although it is outside the scope of this paper we also give numerical results for the iterative solution of the equation for the stream-function. These results underline the relevance of solving (6.2) instead of (6.3). The solution technique we have used is Bi-CGSTAB in combination with an ILU-preconditioner of $rL-C$. The system is solved to about the same precision (for the stream-function) as is achieved if first the equation for the vorticity is solved with one of the methods described above. The iterative solution of equation (6.3) requires about 1000 Bi-CGSTAB iterations.

If we take the number of Bi-CGSTAB iterations as a measure for the amount of work we may conclude that the relaxed inner-outer schemes for (6.2) are much more efficient than Bi-CGSTAB for (6.3). Less than 200 Bi-CGSTAB iterations are required for the relaxed nested schemes for (6.2), while 5 times as many Bi-CGSTAB iterations are needed for solving (6.3). The comparison gives only an indication. In reality it will be even more favorable for (6.2), since by counting the number of Bi-CGSTAB iterations we neglected the overhead for GMRESR (or FGMRES), and the matrix–vector multiplications for solving (6.3) are more expensive.

7. Conclusions

In this paper we have analyzed strategies for controlling the accuracy of approximate matrix–vector products in the context of nested Krylov methods. In a nested Krylov methods the number of expensive accurate matrix–vector in the outer loop is reduced by applying a Krylov method, that uses cheap matrix–vector products with low accuracy as preconditioner. We have shown that an important saving in the work for the matrix–vector products can be achieved using this strategy.

We have analyzed two so called flexible Krylov methods in detail: FGMRES and GMRESR. In both methods, the accuracy of the matrix–vector products can be controlled using a relaxation strategy. We have shown that the two methods, although closely related, behave differently in the case of a near breakdown.
of the associated FOM process. In FGMRES the residual reduction can be less than what can be expected from the reductions obtained in the preconditioning iteration, whereas accuracy problems are not to be anticipated. Conversely, in GMRESR problems can occur with the accuracy of the method in case of a near breakdown, while the residual reduction will be as achieved in the preconditioning iteration.

We have demonstrated the advantages of nested inexact Krylov subspace methods for a Schur complement system that occurs in a model that describes global ocean circulation. The nested schemes are about ten times more efficient than GMRES and five times more efficient than GMRES combined with a relaxation strategy to control the error in the matrix–vector products.

Acknowledgements

We thank the referee for very carefully reading a first version of this paper and for his or her constructive remarks that helped us to significantly improve the clarity of this paper.

We thank Luc Giraud for providing us with a copy of [13] and Henk van der Vorst for bringing [23] to our attention. Part of this research was done during a visit of the first author to CERFACS.

References


