

RECYCLING KRYLOV SUBSPACES FOR SEQUENCES OF LINEAR SYSTEMS*

MICHAEL L. PARKS[†], ERIC DE STURLER[‡], GREG MACKEY[§], DUANE D. JOHNSON[¶],
AND SPANDAN MAITI^{||}

Abstract. Many problems in science and engineering require the solution of a long sequence of slowly changing linear systems. We propose and analyze two methods that significantly reduce the total number of matrix-vector products required to solve all systems. We consider the general case where both the matrix and right-hand side change, and we make no assumptions regarding the change in the right-hand sides. Furthermore, we consider general nonsingular matrices, and we do not assume that all matrices are pairwise close or that the sequence of matrices converges to a particular matrix. Our methods work well under these general assumptions, and hence form a significant advancement with respect to related work in this area. We can reduce the cost of solving subsequent systems in the sequence by recycling selected subspaces generated for previous systems. We consider two approaches that allow for the continuous improvement of the recycled subspace at low cost. We consider both Hermitian and non-Hermitian problems, and we analyze our algorithms both theoretically and numerically to illustrate the effects of subspace recycling. We also demonstrate the effectiveness of our algorithms for a range of applications from computational mechanics, materials science, and computational physics.

Key words. sequence of linear systems, linear solvers, Krylov methods, truncation, restarting, Krylov subspace recycling, iterative methods

AMS subject classifications. 65F10, 65N22

DOI. 10.1137/040607277

1. Introduction. We consider the solution of a long sequence of general linear systems

$$(1.1) \quad A^{(i)}x^{(i)} = b^{(i)}, \quad i = 1, 2, \dots,$$

where the matrix $A^{(i)} \in \mathbb{C}^{n \times n}$ and right-hand side $b^{(i)} \in \mathbb{C}^n$ change from one system to the next, and the systems are typically not available simultaneously. Such sequences arise in many problems.

*Received by the editors April 24, 2004; accepted for publication (in revised form) February 16, 2006; published electronically October 6, 2006.

<http://www.siam.org/journals/sisc/28-5/60727.html>

[†]Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, IL 61801, and Computational Mathematics and Algorithms, Sandia National Laboratories, P.O. Box 5800, MS 1110, Albuquerque, NM 87185 (mlparks@sandia.gov). Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy under contract DE-AC04-94-AL85000. The work of this author was supported by the Computational Science and Engineering program at UIUC through two CSE fellowships.

[‡]Department of Mathematics, Virginia Tech, Blacksburg, VA 24061-0123 (sturler@vt.edu). The work of this author was supported by the Materials Computation Center (UIUC) through grants NSF DMR 99-76550 and DMR-0325939, and by the Center for Simulation of Advanced Rockets (UIUC) through grant DOE LLNL B341494.

[§]Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, IL 61801 (currently gemacke@sandia.gov). The work of this author was supported by the Center for Simulation of Advanced Rockets through grant DOE LLNL B341494.

[¶]Department of Materials Science, University of Illinois at Urbana-Champaign, Urbana, IL 61801 (duanej@uiuc.edu). The work of this author was supported through the Materials Computation Center through grant NSF DMR 99-76550.

^{||}Department of Aerospace Engineering, University of Illinois at Urbana-Champaign, Urbana, IL 61801 (maiti@uiuc.edu).

One important class of applications that we consider occurs in modeling fatigue and fracture via finite element analysis. These analyses use dynamic loading, requiring many loading steps, and rely on implicit solvers [13]. Generally, several thousand loading increments, each corresponding to a linear system, are required to resolve the fracture progression. The matrix and right-hand side, at each loading step, depend on the previous solution, so that only one linear system is available at a time. Further, the solution vectors are updates to a displacement vector and are therefore uncorrelated. Although the change from one linear system to the next is small, the cumulative change after many loading increments is significant. Rather than discarding the Krylov space generated when solving a linear system, we judiciously select a subspace and use it to reduce the number of iterations for solving the next system. We refer to this process as *Krylov subspace recycling*. Clearly, similar sequences of linear systems arise from other (nonlinear) time-dependent applications. Another important source of such sequences of linear systems are Newton or Broyden-type methods for solving nonlinear equations and optimization problems.

We consider methods for the solution of sequences of general matrices, and do not assume that all matrices are pairwise close or that the sequence of matrices converges to a particular matrix. In addition, we make no assumptions on the right-hand-side vectors. A method that is effective under these assumptions must satisfy a number of properties. First, the method must be able to identify and converge to an effective subspace for recycling (the *recycle space*) in a reasonable number of iterations, and it must be able to converge to an effective recycle space over the solution of multiple linear systems. Otherwise, a good recycle space may never be found for a sequence of changing matrices. Second, for efficiency a significant convergence improvement for the linear solver must be obtained with a relatively small recycle space. Third, the method must be able to converge quickly to an effective *perturbed* recycle space for an updated matrix, and it must provide an inexpensive mechanism for regularly updating the recycle space to reflect the changes in the linear systems. As we will show, our proposed methods satisfy these properties, and they are effective under these general assumptions. As such, they form a significant advancement with respect to related work in this area.

We consider two approaches for the solution of (1.1) which are related to two existing truncated and restarted solvers. These solvers, GMRES-DR [21] and GCROT [7], were developed for solving single linear systems; both recycle a judiciously selected subspace between restarts to maintain good convergence. In the following, we define a *truncation* in the sense used by GCROT, wherein the iteration is restarted with a selected subspace and each new vector generated is orthogonalized against this subspace. We define a *cycle* as the computation between truncations or restarts. The first approach is to recycle an approximate invariant subspace and use it for deflation, following the GMRES-DR method. Since GMRES-DR cannot be adapted for recycling, we propose a more general method, GCRO-DR. The work by Rey and Risler [25, 26] has a similar motivation as that for GMRES-DR; see below. However, our implementation is cheaper, more effective, and more adaptive. An alternative idea is to recycle a subspace that minimizes the loss of orthogonality with the Krylov subspace from the previous system [7]. In either case, the recycled subspace will be utilized by first minimizing the residual over this subspace and then maintaining orthogonality with the image of this space in the Arnoldi recurrence. We will show that subspaces that are useful to retain for a subsequent cycle when solving a single linear system are also useful for the next linear system in a sequence if the matrix does

not change significantly. The proposed methods can be tuned to recycle a variety of subspaces based on computed information from the matrix, background knowledge of the application, and other information; this is discussed in [15]. However, for a first evaluation, it is reasonable to analyze the effectiveness of these existing methods appropriately modified for solving (1.1). The application of these methods to sequences of linear systems where both matrix and right-hand side change is new and provides a significant advance on related methods discussed below. Our proposed methods offer an efficient mechanism for continuous updating of the recycled subspace. As a result, they quickly adapt to gradual changes in matrices. Further, they make few assumptions on the linear systems and are applicable to general matrices. In addition, we provide theoretical motivation and careful experimental analysis of these methods. Our results show that significant convergence improvements are obtained using recycled subspaces of a small dimension. The application of an early version of GCROT to multiple right-hand sides was briefly discussed in [5]. The tuning of subspace recycling for diffuse optical tomography, leading to further convergence improvements, is discussed and relevant theory presented in [15].

We discuss the basic derivation of our methods from Morgan's GMRES-DR and de Sturler's GCROT in section 2. We modify GCROT to recycle subspaces between linear systems. GMRES-DR cannot be modified to do this, so we introduce GCRO-DR, a more general variant of GMRES-DR, capable of Krylov subspace recycling. In section 3, we discuss two important theorems and their consequences to explain why GCRO-DR satisfies the three essential properties for an effective recycling method. A similar convergence theory for GCROT is a topic of future research. This analysis is complemented by careful experimentation in section 4, where we provide an extensive experimental analysis to illustrate the behavior of both proposed methods and how well they satisfy the required properties. Particular examples from section 4.4 suggest that further improvements are possible in the subspace selection process, which is a subject of future research. Finally, we discuss conclusions and future work in section 5.

To conclude this section, we discuss a number of related approaches. For the Hermitian positive definite case, Rey and Risler have proposed reducing the effective condition number by either explicitly retaining all converged Ritz vectors from previous CG iterations or implicitly approximating the dominant invariant subspace by retaining all previously generated complete Krylov spaces [25, 26]. Moreover, both approaches use full recurrences, so the CG iteration is really a FOM (full orthogonalization method) iteration [27]. Clearly, both in memory and floating point operations, these methods are very expensive. This drawback is somewhat alleviated as the methods are presented in the context of the finite element tearing and interconnecting (FETI) method [10], which operates on a reduced-size problem. However, for general problems the approach would appear impractical. Furthermore, both approaches lack the possibility to gradually improve the recycle space. Instead, the recycle space is computed once and periodically replaced by starting from scratch. There is no effective mechanism for adapting this space for a sequence of linear systems that change gradually but substantially over many steps. For this reason, the authors make the explicit assumption that all matrices remain close. Although the approaches are motivated by well-known properties of CG, no theoretical or quantitative analysis is provided of the quality of the approximate invariant subspaces or the convergence rate for CG with recycling. Finally, the reported reductions in the number of iterations are modest at about 20%, even though large recycle spaces are used (e.g., of dimension 100 in [25]). In [26] the authors report about a 20% reduction in CPU-time with respect to CG with a full recurrence.

Another interesting approach for solving a sequence, or better a collection, of linear systems was proposed by Chan and Ng [3] developing two related Galerkin projection methods. Unfortunately, these methods require all systems to be available simultaneously, or at least the right-hand sides. They do not recycle subspaces that arise in the iteration, but instead use all vectors arising in the iteration immediately for all systems. Therefore, they focus on situations where all the matrices are close. However, this is generally not the case for the problems we target. Although the matrices change slowly, the cumulative change over many steps is usually significant.

Solving a sequence of linear systems where the matrix is fixed is a special case of (1.1). When all right-hand sides are available simultaneously, block methods are often suitable, such as block CG [22], block GMRES [37], and the family of block EN-like methods [39]. However, block methods do not generalize to the case (1.1). If only one right-hand side is available at a time, the method of Fischer [11], the deflated CG method [29], or the hybrid method of Simoncini and Gallopoulos [30] may be employed. Fischer's method looks for a starting vector in the space spanned by the previous solution vectors in the sequence, which is helpful if the solution vectors are correlated. The method does not maintain orthogonality to this subspace, and so no further speed-up is obtained. In deflated CG, only a small number of the initial Lanczos vectors for every system is used to update the approximate invariant subspace. This is efficient in computation and memory use, but the convergence to an invariant subspace is slow. Hence, the improvement in iterations is modest. The hybrid method of Simoncini and Gallopoulos is most effective when the right-hand sides share common spectral information.

2. Recycling Krylov methods. Restarting GMRES [28] may lead to poor convergence and even stagnation. Therefore, recent research has focused on truncated methods that improve convergence by retaining a judiciously selected subspace between cycles. A taxonomy of popular choices is given in [9]. In this section, we discuss two choices and solvers implementing them. We then modify those solvers to recycle subspaces between linear systems. Although these linear solvers are not new, their extension to sequences of linear systems has not been discussed and analyzed before.

One strategy for subspace selection was proposed in [7] for the GCROT method. We discuss this approach and its modification for solving (1.1) in section 2.2.

We discuss Morgan's GMRES-DR, which retains an approximate invariant subspace between cycles, in section 2.3. In particular, it focuses on removing the eigenvalues of smallest magnitude by recycling an approximate invariant subspace associated with those eigenvalues. Note that GMRES-DR must use only harmonic Ritz vectors, and that it cannot be modified for Krylov subspace recycling even when the matrix does not change. Therefore, we combine ideas from GCRO [6] and GMRES-DR to produce a new linear solver, GCRO-DR, which is suitable for the solution of individual linear systems as well as sequences of them, and is more flexible than GMRES-DR. We discuss GCRO-DR in section 2.4.

2.1. Definitions. The Arnoldi recurrence in GMRES leads to the following relation, which we denote as the Arnoldi relation:

$$(2.1) \quad AV_m = V_{m+1}\underline{H}_m,$$

where $V_m \in \mathbb{C}^{n \times m}$ and $\underline{H}_m \in \mathbb{C}^{(m+1) \times m}$ is upper Hessenberg. Let $H_m \in \mathbb{C}^{m \times m}$ denote the first m rows of \underline{H}_m .

For any subspace $S \subseteq \mathbb{C}^n$, $y \in S$ is a Ritz vector of A with Ritz value θ if

$$(2.2) \quad Ay - \theta y \perp w \quad \forall w \in S.$$

Frequently, we choose $S = K^{(j)}(A, r)$, the j th Krylov subspace associated with the matrix A and the starting vector r . In this case the eigenvalues of H_m are the Ritz values of A . Ritz values tend to approximate the extremal eigenvalues of A well, but can give poor approximations to the interior eigenvalues. Likewise, the Ritz values of A^{-1} tend to approximate the interior eigenvalues of A . We define harmonic Ritz values as the Ritz values of A^{-1} with respect to the space AS ,

$$(2.3) \quad A^{-1}\tilde{y} - \tilde{\mu}\tilde{y} \perp w \quad \forall w \in AS,$$

where again $S = K^{(j)}(A, r)$ and $\tilde{y} \in AS$. We call $\tilde{\theta} = 1/\tilde{\mu}$ a harmonic Ritz value. In this case, we have approximated the eigenvalues of A^{-1} , but using a Krylov space generated with A . In GCRO-DR, we construct harmonic Ritz vectors using a modified operator rather than A .

2.2. GCROT. GCROT is a truncated minimum residual Krylov method that retains a subspace between cycles such that the loss of orthogonality with respect to the discarded space is minimized [7]. This process is called *optimal truncation*. We discuss optimal truncation in the context of restarted GMRES, although it can be described in more general terms, independently of any specific linear solver [7, 18]. Consider solving $Ax = b$ with initial residual r_0 . At the end of the first cycle of GMRES, starting with $v_1 = r_0/\|r_0\|_2$, we have the Arnoldi relation (2.1).

Let r_1 denote the residual vector after m iterations. Consider some iteration $s < m$. For s iterations of GMRES, we have the Arnoldi relation

$$(2.4) \quad AV_s = V_{s+1}H_s.$$

Let r denote the residual after s iterations. Now suppose that we had restarted after iteration s , with initial residual r , and made $m - s$ iterations, yielding residual r_2 . The optimal residual after m iterations is r_1 . At best, we may have $\|r_2\|_2 = \|r_1\|_2$, but in general, $\|r_2\|_2 > \|r_1\|_2$, because GMRES restarted after iteration s ignores orthogonality to the Krylov subspace $AK^{(s)}(A, r_0)$. The deviation from optimality incurred by restarting after iteration s is $e = r_2 - r_1$, which we call the *residual error*. The residual error e depends on the *principal angles* [12, pp. 603–604] between the subspaces $AK^{(s)}(A, r_0)$ and $AK^{(m-s)}(A, r)$. Instead of completely discarding the space $AK^{(s)}(A, r_0)$, suppose we had maintained orthogonality to a k -dimensional subspace ($k < s$) of $AK^{(s)}(A, r_0)$ for the remaining $m - s$ iterations to produce a new residual vector r_3 . If we chose our k -dimensional subspace of $AK^{(s)}(A, r_0)$ to correspond to the k largest principal angles, we would minimize the norm of the new residual error $\|e'\|_2 = \|r_1 - r_3\|_2$. This process is what is meant by optimal truncation. Since the Krylov space generated with r contained vectors close to the recycled subspace, this is likely to happen again after restarting with r_1 . Therefore, we retain the selected k -dimensional subspace for the next cycle.

GCROT maintains matrices $U_k, C_k \in \mathbb{C}^{n \times k}$ satisfying the relations

$$(2.5) \quad AU_k = C_k,$$

$$(2.6) \quad C_k^H C_k = I_k.$$

The minimum residual solution over $\text{range}(U_k)$ is known from the previous cycle. In the following cycle, we carry out the Arnoldi recurrence while maintaining orthogonality to C_k . This corresponds to an Arnoldi recurrence with the operator $(I - C_k C_k^H)A$.

Then, we compute the update to the solution as in GMRES, taking the singularity of the operator into account using the relation $A^{-1}(I - C_k C_k^H)A = I - U_k C_k^H A$ [6]. The correction to the solution vector and other vectors selected via optimal truncation of the Krylov subspace are appended to U_k , and then U_k and C_k are updated such that (2.5)–(2.6) again hold. At the end of each cycle, only the matrices U_k and C_k are carried over to the next cycle. Each cycle of GCROT requires approximately $m - k$ matrix-vector products and $O(nm^2 + nkm)$ other floating point operations. For details, see [7].

GCROT can be modified to solve (1.1) by carrying over U_k from the i th system to the $(i + 1)$ st system. We have the relation $A^{(i)}U_k = C_k$. We modify U_k and C_k to satisfy (2.5)–(2.6) with respect to $A^{(i+1)}$ as follows:

$$(2.7a) \quad [Q, R] = \text{reduced QR decomposition of } A^{(i+1)}U_k^{old},$$

$$(2.7b) \quad C_k^{new} = Q,$$

$$(2.7c) \quad U_k^{new} = U_k^{old}R^{-1}.^1$$

Now, $A^{(i+1)}U_k^{new} = C_k^{new}$, and we can proceed with GCROT on the $(i + 1)$ st linear system. Note that in many cases computing $A^{(i+1)}U_k^{old} = C_k^{old} + \Delta A^{(i)}U_k^{old}$ is *much* cheaper than k matrix-vector products, because $\Delta A^{(i)}$ is considerably sparser than $A^{(i)}$ or has a special structure. See our example problem in section 4.1. Moreover, even if we were to compute $A^{(i+1)}U_k^{old}$ directly, this can be faster than k separate matrix-vector multiplications [8]. So long as $A^{(i+1)}$ has not changed significantly from $A^{(i)}$, the use of U_k^{new} should accelerate the solution of the $(i + 1)$ st linear system.

2.3. GMRES-DR. GMRES-DR [21] recycles an approximate invariant subspace to deflate eigenvalues of smallest magnitude. Deflating these eigenvalues can greatly improve convergence in certain circumstances.

In each cycle, GMRES-DR carries forward k harmonic Ritz vectors $\tilde{Y}_k \in \mathbb{C}^{n \times k}$ computed at the end of the previous cycle. For the first cycle, the harmonic Ritz vectors can be computed from H_m in (2.1). It can be shown that these harmonic Ritz vectors fit naturally into a Krylov subspace [20]. In each cycle, GMRES-DR proceeds by first orthogonalizing \tilde{Y}_k to give $\tilde{\Upsilon}_k$. GMRES-DR then carries out the Arnoldi recurrence for $m - k$ iterations while maintaining orthogonality to $\tilde{\Upsilon}_k$. This gives the Arnoldi-like relation

$$(2.8) \quad A[\tilde{\Upsilon}_k \quad V_{m-k}] = [\tilde{\Upsilon}_k \quad V_{m-k+1}]\underline{H}_m,$$

where \underline{H}_m is upper Hessenberg, except for a leading dense $(k + 1) \times (k + 1)$ submatrix. GMRES-DR updates the solution and residual as in GMRES. It then computes the harmonic Ritz vectors associated with the k smallest harmonic Ritz values using (2.8), and finally restarts with those vectors.

GMRES-DR cannot be used to solve (1.1) directly, even if the matrix is fixed. The harmonic Ritz vectors of A in \tilde{Y}_k do not form a Krylov subspace for another matrix or even just another starting vector. These and other reasons lead us to develop GCRO-DR, a generalization of GMRES-DR capable of solving (1.1).

2.4. GCRO-DR. We introduce a new Krylov method that uses recycling. We call this method GCRO-DR because it uses deflated restarting within the framework of GCRO [6]. The method is a generalization of GMRES-DR to solve (1.1). GCRO-DR is more flexible because *any* subspace may be recycled for subsequent cycles or

¹For efficiency, U_k^{new} need not be computed explicitly.

linear systems. In the pseudocode given in the appendix, the harmonic Ritz vectors corresponding to the harmonic Ritz values of smallest magnitude have been chosen. However, any combination of k vectors may be selected. An interesting possibility would be to select a few *harmonic Ritz vectors* corresponding to the harmonic Ritz values of smallest magnitude, and a few *Ritz vectors* corresponding to the Ritz values of largest magnitude. This would allow simultaneous deflation of eigenvalues of both smallest and largest magnitude using the better approximation for each.

When solving a single linear system, GCRO-DR and GMRES-DR are algebraically equivalent. The primary advantage of GCRO-DR is its capability for solving sequences of linear systems.

GCRO-DR is a combination of GMRES-DR and GCRO. Suppose that we have solved the i th system of (1.1) with GCRO-DR, and we retain k approximate eigenvectors, $\tilde{Y}_k = [\tilde{y}_1, \tilde{y}_2, \dots, \tilde{y}_k]$. Then, GCRO-DR computes matrices $U_k, C_k \in \mathbb{C}^{n \times k}$ from \tilde{Y}_k and $A^{(i+1)}$ such that $A^{(i+1)}U_k = C_k$ and $C_k^H C_k = I_k$, in the same manner as in (2.7). In the remainder of this section we drop the superscript in $A^{(i+1)}$ for notational convenience.

We find the optimal solution over the subspace $\text{range}(U_k)$ as $x = x_0 + U_k C_k^H r_0$, and set $r = r_0 - C_k C_k^H r_0$ and $v_1 = r / \|r\|_2$. Next, we generate a Krylov space of dimension $m - k + 1$ with $(I - C_k C_k^H)A$, which produces the Arnoldi relation

$$(2.9) \quad (I - C_k C_k^H)A V_{m-k} = V_{m-k+1} \underline{H}_{m-k}.$$

Since $V_{m-k+1} \perp C_k$, we have

$$(2.10) \quad A[U_k \ V_{m-k}] = [C_k \ V_{m-k+1}] \begin{bmatrix} I_k & B_k \\ 0 & \underline{H}_{m-k} \end{bmatrix},$$

where $B_k = C_k^H A V_{m-k}$. To reduce unnecessary ill-conditioning of the rightmost matrix in (2.10) we proceed as follows. We compute the diagonal matrix D_k such that $\tilde{U}_k = U_k D_k$ has unit columns, and we define

$$\hat{V}_m = [\tilde{U}_k \ V_{m-k}], \quad \hat{W}_{m+1} = [C_k \ V_{m-k+1}], \quad \underline{G}_m = \begin{bmatrix} D_k & B_k \\ 0 & \underline{H}_{m-k} \end{bmatrix}.$$

We rewrite (2.10) as

$$(2.11) \quad A \hat{V}_m = \hat{W}_{m+1} \underline{G}_m,$$

where the columns of \hat{V}_m and \hat{W}_{m+1} have unit norm. Note that $\underline{G}_m = \hat{W}_{m+1}^H A \hat{V}_m$ is upper Hessenberg, with D_k diagonal. The columns of \hat{W}_{m+1} are orthogonal, but this is not true for the columns of \hat{V}_m .

At the end of each cycle, GCRO-DR solves the minimization problem

$$(2.12) \quad t = \arg \min_{z \in \text{range}(\hat{V}_m)} \|r - Az\|_2,$$

which reduces to the $(m+1) \times m$ least squares problem

$$(2.13) \quad \underline{G}_m y \approx \hat{W}_{m+1}^H r = \|r\|_2 e_{k+1},$$

with $t = \hat{V}_m y$. The residual and solution are given by

$$(2.14) \quad r = r - A \hat{V}_m y = r - \hat{W}_{m+1} \underline{G}_m y,$$

$$(2.15) \quad x = x + \hat{V}_m y.$$

Next, the method solves the generalized eigenvalue problem

$$(2.16) \quad \underline{G}_m^H \underline{G}_m \tilde{z}_i = \tilde{\theta}_i \underline{G}_m^H \widehat{W}_{m+1}^H \widehat{V}_m \tilde{z}_i,$$

derived from (2.3), and recovers the harmonic Ritz vectors as $\tilde{y}_i = \widehat{V}_m \tilde{z}_i$. In general, (2.16) will have complex eigenvalues. When storing the harmonic Ritz vectors in GCRO-DR, we use a real representation of a complex conjugate pair. To ensure that we retain both complex conjugate harmonic Ritz vectors associated with a selected eigenvalue, it is sometimes necessary to store $k+1$ vectors rather than k vectors per cycle.

GCRO-DR and GMRES-DR have about the same computational cost per cycle. In particular, they have the same number of matrix-vector products and orthogonalizations per cycle. GCRO-DR stores k additional vectors. If the new U_k is computed explicitly (which is not necessary), GCRO-DR has a modest additional computational cost of about $nk^2/2$. Given U_k and C_k , generating (2.11) with GCRO-DR(m, k) requires approximately $2kn(1+k)$ fewer floating point operations than generating (2.1) with GMRES(m), although GMRES(m) stores k fewer vectors. The number of dot-products and vector updates per cycle is of the same order for GCRO-DR(m, k) and GMRES(m); the cost savings in GCRO-DR(m, k) arise because (2.5) and (2.6) are already satisfied.

3. Convergence analysis for deflation-based Krylov subspace recycling.

Recent work on the convergence of GMRES [31] together with the theory on invariant subspaces and their perturbations [34] provides a good framework for analyzing the GCRO-DR method. Unfortunately, a similar convergence theory for the GCROT method is still lacking. However, in section 4 we show by numerical experiment that, regarding recycling, GCROT shares many of the properties of GCRO-DR. A full theoretical analysis of GCRO-DR is beyond the scope of the present paper; instead we discuss two main theoretical results and their implications and demonstrate these numerically in section 4. For more details on these theoretical results we refer to [15, 23, 24].

The first result concerns the convergence of GCRO-DR; see [23, 24]. We show that the recycle space need not approximate an invariant subspace accurately to improve the rate of convergence significantly.

Let \mathcal{Q} be an ℓ -dimensional invariant subspace of A , and let $\mathcal{C} = \text{range}(C_k)$ be a k -dimensional space ($k \geq \ell$) selected to approximate \mathcal{Q} , where C_k is defined in section 2.4. We define $\Pi_{\mathcal{C}}$ to be the orthogonal projector onto \mathcal{C} , and we define $\Pi_{\mathcal{Q}}$ similarly. Furthermore, we define $P_{\mathcal{Q}}$ to be the spectral projector onto \mathcal{Q} . Finally, we define the *one-sided distance* from the subspace \mathcal{Q} to the subspace \mathcal{C} as

$$(3.1) \quad \delta(\mathcal{Q}, \mathcal{C}) \equiv \|(I - \Pi_{\mathcal{C}})\Pi_{\mathcal{Q}}\|_2,$$

which is equal to the sine of the largest principal angle between \mathcal{Q} and \mathcal{C} [1]. This means that any unit vector in \mathcal{Q} has a component of at most length δ orthogonal to \mathcal{C} .

THEOREM 3.1. *Given a space \mathcal{C} , let $\mathcal{V} = \text{range}(V_{m-k+1}\underline{H}_{m-k})$ be the $(m-k)$ -dimensional Krylov subspace generated by GCRO-DR as in (2.9). Let $r_0 \in \mathbb{C}^n$, and let $r_1 = (I - \Pi_{\mathcal{C}})r_0$. Then, for each \mathcal{Q} such that $\delta(\mathcal{Q}, \mathcal{C}) < 1$,*

$$\begin{aligned} \min_{d_1 \in \mathcal{V} \oplus \mathcal{C}} \|r_0 - d_1\|_2 &\leq \min_{d_2 \in (I - P_{\mathcal{Q}})\mathcal{V}} \|(I - P_{\mathcal{Q}})r_1 - d_2\|_2 \\ &\quad + \frac{\gamma}{1 - \delta} \|P_{\mathcal{Q}}\|_2 \cdot \|(I - \Pi_{\mathcal{V}})r_1\|_2, \end{aligned}$$

where $\gamma = \|(I - \Pi_{\mathcal{C}})P_{\mathcal{Q}}\|_2$.

If, in addition, A is Hermitian, then we have

$$\begin{aligned} \min_{d_1 \in \mathcal{V} \oplus \mathcal{C}} \|r_0 - d_1\|_2 &\leq \min_{d_2 \in (I - \Pi_{\mathcal{Q}})\mathcal{V}} \|(I - \Pi_{\mathcal{Q}})r_1 - d_2\|_2 \\ &\quad + \frac{\delta}{1 - \delta} \cdot \|(I - \Pi_{\mathcal{V}})r_1\|_2. \end{aligned}$$

Proof. For the proof, see [23, Chapter 3] or [24]. \square

Theorem 3.1 was inspired by a related theorem in [31] used to explain superlinear convergence in GMRES. In the two bounds above, the left-hand side represents the residual norm after $m - k$ iterations of GCRO-DR with the recycled subspace \mathcal{C} . On the right-hand sides, the first term represents the convergence of a deflated problem where all components in the subspace \mathcal{Q} have been removed, which typically leads to an improved rate of convergence [21, 31, 36]. The second term in the right-hand sides represents a constant times the residual of $m - k$ iterations of GCRO-DR, solving for r_1 . If the recycle space \mathcal{C} contains an invariant subspace \mathcal{Q} , then $\delta = \gamma = 0$ for this \mathcal{Q} , and GCRO-DR converges at least as fast as the deflated problem. In our numerical experiments we demonstrate that the method fairly quickly gets to values of $\delta = O(10^{-2})$. In that case, we still obtain the convergence rate of the deflated problem, so long as $\|P_{\mathcal{Q}}\|_2$ is not large in the non-Hermitian case. Notice that for $\delta = O(10^{-2})$ the invariant subspace \mathcal{Q} is not approximated very accurately, and that such values of δ are relatively easily obtained. Finally, we point out that for a given subspace \mathcal{C} , Theorem 3.1 is applicable to any invariant subspace \mathcal{Q} such that $\delta(\mathcal{Q}, \mathcal{C}) < 1$. Hence, the sharpest bound for any \mathcal{Q} applies, and the result appears to be fairly insensitive to conditioning issues of invariant subspaces for non-Hermitian A .

The second result concerns the perturbation of invariant subspaces associated with the smallest eigenvalues when the change in the matrix is concentrated in an invariant subspace corresponding to large eigenvalues. When the magnitude of the change is smaller than the gap between smallest and large eigenvalues, then the invariant subspace associated with the smallest eigenvalues is not significantly altered. As we aim to recycle exactly this subspace, this is a desirable property.

For simplicity we deal specifically with a Hermitian positive definite matrix A and a corresponding Hermitian perturbation E , as in our main numerical example in section 4. Following the discussion in [15], let A have the eigendecomposition

$$(3.2) \quad A = [Q_1 \ Q_2 \ Q_3] \operatorname{diag}(\Lambda_1, \Lambda_2, \Lambda_3) [Q_1 \ Q_2 \ Q_3]^H,$$

where $Q = [Q_1 \ Q_2 \ Q_3]$ is an orthogonal matrix, $\Lambda_1 = \operatorname{diag}(\lambda_1^{(1)}, \dots, \lambda_{j_1}^{(1)})$, and Λ_2 and Λ_3 are defined analogously. Furthermore, let

$$\lambda_1^{(1)} \leq \dots \leq \lambda_{j_1}^{(1)} < \lambda_1^{(2)} \leq \dots \leq \lambda_{j_2}^{(2)} < \lambda_1^{(3)} \leq \dots \leq \lambda_{j_3}^{(3)}.$$

Now we consider the change in the invariant subspace $\operatorname{range}(Q_1)$ under a symmetric perturbation E of A . Let $\theta_1(\cdot, \cdot)$ denote the largest canonical angle between two spaces. We do not require that $\|E\|_F$ be small, but we assume that the projection of E onto the subspace $\operatorname{range}([Q_1 \ Q_2])$ is small. We assume that $\|[Q_1 \ Q_2]^H E\|_F \leq \varepsilon$ and that ε is small relative to $\lambda_1^{(2)} - \lambda_{j_1}^{(1)}$. We also assume that $\eta \equiv \|Q_3^H E\|_F$ is small relative to $\lambda_1^{(3)} - \lambda_{j_1}^{(1)}$. Note that we do not need to assume that $\lambda_1^{(2)} - \lambda_{j_1}^{(1)}$ is large.

Also, let

$$\mu \equiv \min(\lambda_1^{(2)} - \varepsilon, \lambda_1^{(3)} - \eta) - 2\varepsilon - (\lambda_{j1}^{(1)} + \varepsilon) > 2\varepsilon,$$

$$\hat{\mu} \equiv \mu \left(1 - \frac{2\varepsilon^2}{\mu^2} \right) + \lambda_{j1}^{(1)} + \varepsilon.$$

THEOREM 3.2. *Let A be Hermitian positive definite and have the eigendecomposition given in (3.2), and let E , ε , η , μ , and $\hat{\mu}$ be defined as above. Then there exists a matrix \hat{Q}_1 conforming to Q_1 such that $\text{range}(\hat{Q}_1)$ is a simple invariant subspace of $A + E$, and*

$$\tan \theta_1 \left(\text{range}(Q_1), \text{range}(\hat{Q}_1) \right) \leq \frac{\varepsilon}{\hat{\mu}}.$$

Proof. For the proof, see [15]. \square

A similar bound holds for the perturbation of the eigenvalues associated with Q_1 .

In relation to Theorem 3.1 and GCRO-DR, Q_1 corresponds to \mathcal{Q} , whereas Q_2 and Q_3 can be chosen to fit the theorem. We need a specialized perturbation result of this kind, because in general the changes in the matrices are too large to show by standard perturbation theory that the invariant subspaces of interest remain intact. This perturbation theorem is of general importance, as there are many applications that involve a sequence of problems that undergo small local changes. In a problem like crack propagation we do not expect the smooth global modes associated with the smallest eigenvalues to change much from one system to the next, but only gradually over many systems as the crack propagates over some distance. We show experimentally that this is the case in section 4. The continual adaptation of the recycle space needs to track only these gradual changes. Another example is given in [15], where we optimize a parameterized medium in a tomography application to fit measured data.

Finally, we note that GCRO-DR uses more or less the Arnoldi method with dense restarting for approximating an invariant subspace [33, 38]. This method generally offers fast convergence for the exterior components of the spectrum. This fast convergence together with Theorems 3.1 and 3.2 indicates that GCRO-DR satisfies the three important properties mentioned in the introduction, especially for problems such as crack propagation.

The perturbation result indicates that if the recycle space provides a reasonable approximation to an invariant subspace, then it will also provide a reasonable approximation to the slightly perturbed invariant subspace of the updated matrix. The convergence result indicates that GCRO-DR will have fast convergence as the method does not require an accurate approximation to the invariant subspace. Moreover, Arnoldi's method with dense restarting will quickly improve the approximation to the invariant subspace of the perturbed matrix. In general, it is not hard to get a reasonable approximation to an invariant subspace corresponding to the outermost eigenvalues [34]. We will demonstrate this behavior of GCRO-DR using numerical experiments in the next section.

4. Test problems and numerical results. We discuss our main example in section 4.1, a problem from fracture mechanics that produces a long sequence of linear systems. The matrices are symmetric positive definite (SPD), and both the matrix and right-hand side change from one system to the next. For this problem we also provide a more detailed experimental analysis of the GCRO-DR and GCROT methods following the theory described in section 3. In addition, we provide results for

three problems that involve real nonsymmetric matrices and complex non-Hermitian matrices. We consider two examples from physics to illustrate the effectiveness of our approach for the case of a fixed matrix. We discuss electronic structure calculations in section 4.2, and a problem from lattice quantum chromodynamics in section 4.3. Finally, in section 4.4, we apply GCROT and GCRO-DR to a simple convection-diffusion problem to evaluate the effects of subspace recycling in the nonsymmetric case, independent from perturbations in the matrix or right-hand side.

In the following sections, $\text{GMRES}(m)$ indicates restarted GMRES with a maximum subspace of dimension m , and $\text{GMRES}(\infty)$ indicates full GMRES. CG refers to the conjugate gradient method. For $\text{GMRES-DR}(m, k)$ and $\text{GCRO-DR}(m, k)$, m is the maximum subspace size, and k is the dimension of the recycle space. For $\text{GCROT}(m, k_{\max}, k_{\min}, s, p_1, p_2)$, m is the maximum subspace size over which we optimize. The maximum number of column vectors stored in U_k and C_k (as described in section 2.2) is k_{\max} . The argument k_{\min} indicates the number of column vectors retained in U_k and C_k after truncation to make room for new vectors. The argument s indicates the dimension of the Krylov subspace from which we select p_1 vectors to place in U_k . We also include in U_k the last p_2 orthogonal basis vectors generated in the Arnoldi process. See [7, 18] for more discussion regarding the choice of parameters. At each restart for GCROT, the GMRES part is run for $m - k_{\min}$ steps.

In comparing restarted GMRES, GCROT, GMRES-DR, and GCRO-DR, we decided to make the solvers minimize over a subspace of the same dimension. An alternative choice would be to provide the same amount of memory to each solver, but we felt that our choice would provide a more informative comparison.

4.1. Fatigue and fracture of engineering components. Research on failure mechanisms (e.g., fatigue and fracture) of engineering components often focuses on modeling complex, nonlinear response. Finite element methods for quasi-static and transient responses over longer time scales generally adopt an implicit formulation. Together with a Newton scheme for the nonlinear equations, such implicit formulations require the solution of linear systems, thousands of times, to accomplish a realistic analysis [13].

We study a sequence of linear systems taken from a finite element code developed by Philippe Geubelle and Spandan Maiti (both Aerospace Engineering, University of Illinois at Urbana-Champaign (UIUC)). In our example, the code simulates crack propagation in a metal plate using so-called *cohesive finite elements*. The plate mesh is shown in Figure 4.1. The problem is symmetric about the x -axis, and the crack propagates exactly along this symmetry axis. The cohesive elements act as nonlinear springs connecting the surfaces that will define the crack location. As the crack propagates, the cohesive elements deform following a nonlinear yield curve and eventually break. The element stiffness is set to zero for a broken cohesive element. These elements are usually inserted dynamically, but that is not the case here. This simulation results in a sequence of sparse, symmetric, positive definite stiffness matrices that change slowly from one system to the next. Each stiffness matrix can be expressed as $A^{(i+1)} = A^{(i)} + \Delta A^{(i)}$. Although $\Delta A^{(i)}$ is considerably more sparse than $A^{(i)}$, it is not low-rank, as the terms in the update $\Delta A^{(i)}$ come from all nonbroken cohesive elements. The other finite elements model linear elasticity and have constant stiffness matrices. The matrices produced in our examples are 3988×3988 , and have a condition number on the order of 10^4 before preconditioning. They have an average of 13.4 nonzero entries per row. Over 2000 linear systems must be solved to capture the fracture progression.

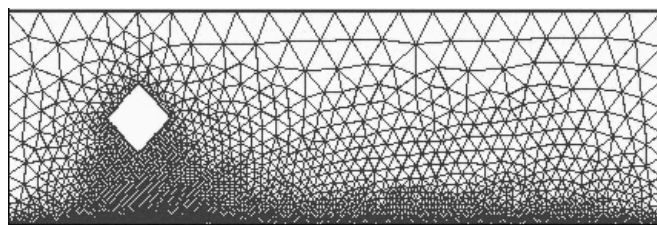
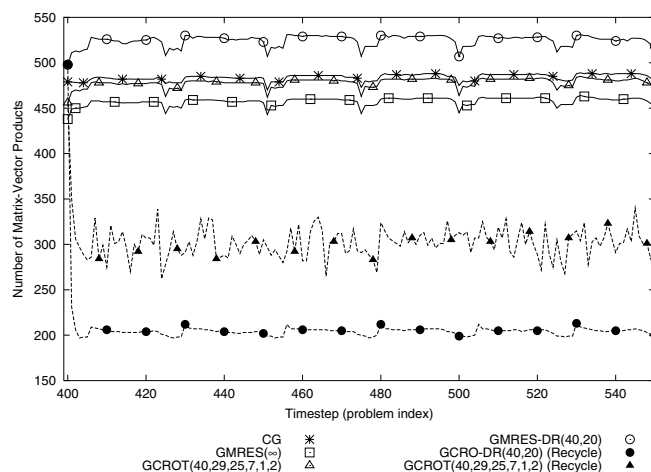


FIG. 4.1. Two-dimensional plate mesh for the crack propagation problem.

FIG. 4.2. Number of matrix-vector products versus timestep for various solvers for the crack propagation problem without preconditioning. All solvers use a recurrence of at most 40 vectors, except GMRES(∞) and CG.

We examine the 151 linear systems 400–550, representing a typical subset of the fracture progression in which many cohesive elements break. We start with a straightforward comparison of GCRO-DR and GCROT with subspace recycling with CG, restarted GMRES, GMRES(∞), GMRES-DR, and GCROT without subspace recycling. We always start with a zero initial guess, since we solve for the incremental displacement associated with the loading increment. We give both preconditioned and nonpreconditioned convergence results. All solvers are required to reduce the relative residual to 1.0×10^{-10} . In Figure 4.2 we give the number of matrix-vector products needed to solve each of these systems without preconditioning for GMRES(∞), CG, GMRES-DR(40,20), GCRO-DR(40,20), and GCROT(40,34,30,5,1,2), both with and without subspace recycling. Except for GMRES(∞) and CG, all methods in Figure 4.2 minimize over a subspace of dimension 40 in each cycle. GMRES(40) is not shown, because it required too many matrix-vector products. In Figure 4.3 we give results for the same sequence of problems and the same methods but with incomplete Cholesky (IC(0)) preconditioning. A new preconditioner was computed for each matrix, which is not the most efficient approach. The total number of matrix-vector products needed to solve all 151 preconditioned linear systems is given in Table 4.1.

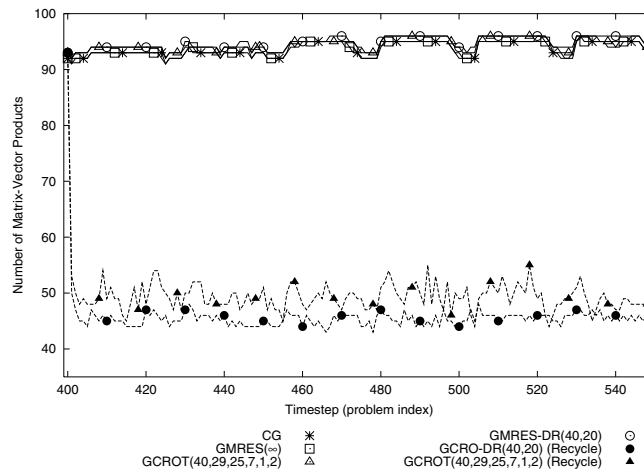


FIG. 4.3. Number of matrix-vector products versus timestep for various solvers for the crack propagation problem with incomplete Cholesky preconditioning. All solvers use a recurrence of at most 40 vectors, except $\text{GMRES}(\infty)$ and CG.

TABLE 4.1

The total number of iterations required to solve 151 consecutive $\text{IC}(0)$ preconditioned linear systems. Only GCRO-DR and GCROT(recycle) exploit subspace recycling.

Method	Matrix-vector products
GMRES(40)	27188
GMRES-DR(40,20)	14305
GCROT(40,34,30,5,1,2)	14277
CG	14162
GMRES(∞)	14142
GCROT(40,34,30,5,1,2) (recycle)	7482
GCRO-DR(40,20) (recycle)	6901

We see in Figure 4.2 that GCRO-DR requires the fewest matrix-vector products, except for the first system in the sequence when no recycle space is available. For the first system, GCROT outperforms GCRO-DR. For all but the first system, GCRO-DR and GCROT with recycling require significantly fewer matrix-vector products than the solvers without subspace recycling. Furthermore, GCROT without recycling and CG show about the same convergence. $\text{GMRES}(\infty)$ outperforms CG, indicating that the convergence of CG is delayed due to effects of finite-precision arithmetic.

For the preconditioned case shown in Figure 4.3, GCRO-DR performs best, with GCROT with subspace recycling a close second. All the other solvers cluster near $\text{GMRES}(\infty)$.

Comparing GMRES-DR and GCRO-DR, we see a significant difference in convergence, even though both methods aim to retain the same approximate invariant subspace. This difference is due solely to subspace recycling. Without the space recycled from previous linear systems, GCRO-DR is algebraically equivalent to GMRES-DR. The results indicate that the invariant subspace associated with the smallest eigenvalues is hard to estimate accurately. GCRO-DR exhibits superior performance, because it does not have to compute that space from scratch for each new linear system. Indeed, we show below that after solving only three linear systems, GCRO-DR has computed a reasonable approximation to the invariant subspace associated with

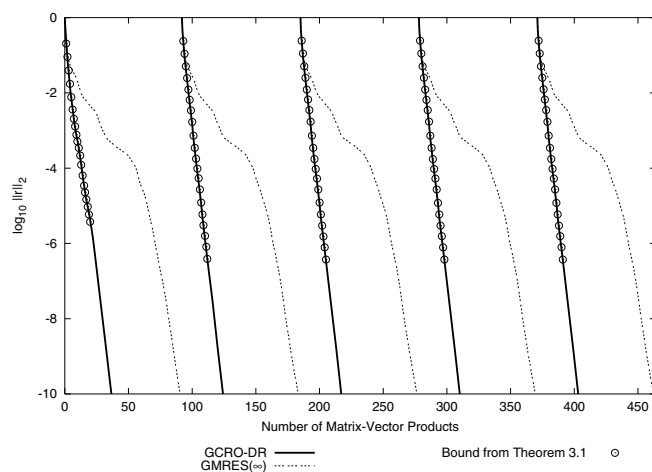


FIG. 4.4. Convergence curves for GCRO-DR on five consecutive linear systems from the preconditioned crack propagation problem, along with the bound described in Theorem 3.1. The bound was computed with \mathcal{Q} the eigenspace corresponding to the 20 smallest eigenvalues of the preconditioned operator. Since the bound is tight and δ is small, GCRO-DR achieved the convergence rate of the deflated problem in Theorem 3.1.

TABLE 4.2

Table of $\delta(\mathcal{Q}, \mathcal{C})$ for GCRO-DR(40, 20) on the preconditioned crack propagation problem, where \mathcal{Q} has been selected as the invariant subspace associated with the 15 eigenvalues of smallest magnitude. The decrease of δ within a column indicates the improvement of the approximation to \mathcal{Q} for a particular system. The small perturbation of the invariant subspace is reflected in the minor increase in δ from the last cycle of one system to before the first of the next system. At each cycle, GCRO-DR updates its approximate invariant subspace.

Linear system	401	402	403	404	405
Before cycle 1	0.9994	0.2380	0.0464	0.0607	0.0794
After cycle 1	0.9180	0.1448	0.0463	0.0606	0.0787
After cycle 2	0.2446	0.0302	0.0416	0.0568	0.0690
After cycle 3	0.2331	0.0302	0.0415	0.0567	0.0684

the fifteen smallest eigenvalues; GMRES-DR cannot do this.

Next, we evaluate some of the properties of GCRO-DR related to the two theoretical results in section 3 and the three properties mentioned in the introduction.

As GCRO-DR approximates an invariant subspace better, it gets closer to the convergence rate of a deflated problem, as described in Theorem 3.1. The convergence curves for five consecutive systems from the preconditioned crack propagation problem are shown in Figure 4.4, along with the bound described in Theorem 3.1. The bound was computed with the invariant subspace, \mathcal{Q} , corresponding to the 20 smallest eigenvalues of the preconditioned matrix. We see that the bound is sharp. Additionally, note that for each problem the initial convergence rate for GCRO-DR is approximately the same as the final convergence rate of GMRES(∞).

In Table 4.2, for the preconditioned example we show how the one-sided distance δ , defined in (3.1), from the invariant subspace associated with the 15 smallest eigenvalues to the recycle space evolves over multiple cycles and multiple linear systems. This table illustrates a number of important issues.

The first system solved in this experiment is system 400, and so system 401 is the first system that starts with a recycle space. For the chosen invariant subspace δ is

quite poor at the start, but δ improves quickly over the next two linear systems, and after cycle 2 for system 402, δ has taken a reasonably small value. So, GCRO-DR is able to compute from scratch increasingly good approximations to invariant subspaces that change slightly from system to system. Note that GMRES-DR starts from scratch for every new system. Therefore, it never obtains a sufficiently good approximation, and hence requires many additional iterations per linear system. Any other method that cannot continually update the approximation to the invariant subspace will suffer from the same problem.

Next, note how δ improves during the solution of a single linear system, whereas the small perturbation of the invariant subspace for the next linear system is reflected in the minor increase in δ from the last cycle of one system to before the first cycle of the next system. This confirms our earlier statement, based on Theorem 3.2, that the invariant subspace associated with the smallest eigenvalues and smooth global modes changes little under small local changes in the model. We have also verified this explicitly by computing the principal angles between invariant subspaces of successive matrices. However, in the next experiment we show that the cumulative change over a larger number of loading steps requires the continual or at least periodic updating of the recycle space to keep the number of iterations small.

Finally, this particular \mathcal{Q} was chosen to illustrate the role and behavior of δ , not to get the best bound from Theorem 3.1. Initially, a smaller invariant subspace might have a much smaller δ and may lead to a tighter bound. The bound holds for any invariant subspace with $\delta < 1$, and so the *smallest* bound is the effective one.

Next, we provide a number of experiments that illustrate how quickly GCRO-DR and GCROT learn and adapt to an updated linear system, and how the convergence rate deteriorates if we stop updating the recycle space.

Figure 4.5 shows the results for three variants of GCRO-DR applied to the non-preconditioned problem. Standard GCRO-DR, as discussed so far, updates the recycle space every cycle, as this update is generally quite cheap. We also show a GCRO-DR variant that runs standard GCRO-DR for the first five linear systems in the sequence, but after that never updates the approximate invariant subspace. This also means that it makes no updates while solving each single system. In this case, convergence deteriorates slowly for the first few linear systems but then more rapidly as the gradual changes in the matrices accumulate. Eventually, the number of iterations per linear system exceeds that of GMRES-DR (cf. Figure 4.2). The third variant of GCRO-DR runs normal GCRO-DR for the first five linear systems in the sequence, then stops updating, except that it runs a standard GCRO-DR every 250th linear system. This frequency was chosen solely for the purpose of demonstration. Clearly, for this problem, the recycled subspace must be updated more frequently for good convergence, say every 25 or 50 systems. Note how, despite the significant deterioration in the number of iterations per system, GCRO-DR recovers the rate of convergence of standard GCRO-DR over the solution of a single system. This shows how quickly the method is able to recover (learn) a sufficiently accurate approximation to the invariant subspace. This example also suggests that the convergence is quite sensitive to the δ values for the relevant invariant subspaces. For this example, the average value of $\|A^{(i)}\|_2$ is 3.1357×10^{10} , and the average value of $\|\Delta A^{(i)}\|_2$ is 2.9076×10^8 . In a relative sense, the change from one matrix to the next is small, but the cumulative change over many linear systems can be quite large. As mentioned before, $\|\Delta A^{(i)}\|_2$ is larger than the smallest eigenvalues. However, the values of δ in Table 4.2 show that the perturbation of the corresponding invariant subspace is quite small, as suggested

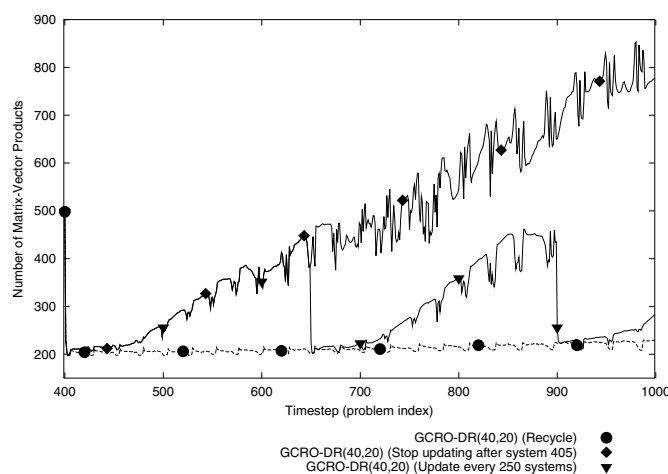


FIG. 4.5. Number of matrix-vector products versus timestep for GCRO-DR(40, 20) with subspace recycling on the nonpreconditioned crack propagation problem. We also show a modified version of GCRO-DR(40, 20) that stops updating the recycled subspace after linear system 405, and a version that stops updating after linear system 405 but does update every 250 systems thereafter. Convergence quickly deteriorates in the latter two cases, showing the importance of updating the approximate invariant subspace regularly.

by the discussion in section 3 regarding small localized changes in the problem and by Theorem 3.2. Figure 4.5 shows that GCRO-DR can adapt to the slow change of the invariant subspace. Finally, it would not be hard to dynamically balance the cost of updating the recycle space with the improved rate of convergence.

In Figure 4.6, we give the results for a similar experiment with GCROT. We show results for standard GCROT with recycling, a GCROT variant that stops updating the recycle space after the first five linear systems, and for a variant that stops updating after the first five linear systems except that it runs a standard GCROT every 100th system. Without the continual updating of the recycle space, the convergence of GCROT degrades more severely than that of GCRO-DR. After each linear system solved by standard GCROT, the number of iterations per system remains small for four or five subsequent systems, but then increases rapidly. This is likely caused by two factors. Since GCROT recycles a subspace that is important for maintaining orthogonality for the current iterations, the recycled space may have a more temporary relevance than an invariant subspace. If we stop updating, then GCROT more or less reduces to restarted GMRES after a number of updates to the matrix when the recycle space is no longer relevant. In addition, after solving a linear system, standard GCROT recycles a subspace that was relevant at the end of that iteration, which may not be the best recycle space for the start of the iteration for the next system. Note that solving a single linear system with standard GCROT restores the convergence rate to that for using standard GCROT for each system. This shows that the method is able to *learn* quickly.

These experiments demonstrate that GCRO-DR and GCROT are able to make the small changes to the recycle space necessitated by small localized changes in the problem. For GCRO-DR this is explained by the perturbation result from Theorem 3.2 in combination with the convergence predicted by Theorem 3.1 for small enough δ (say, $\delta = O(10^{-2})$).

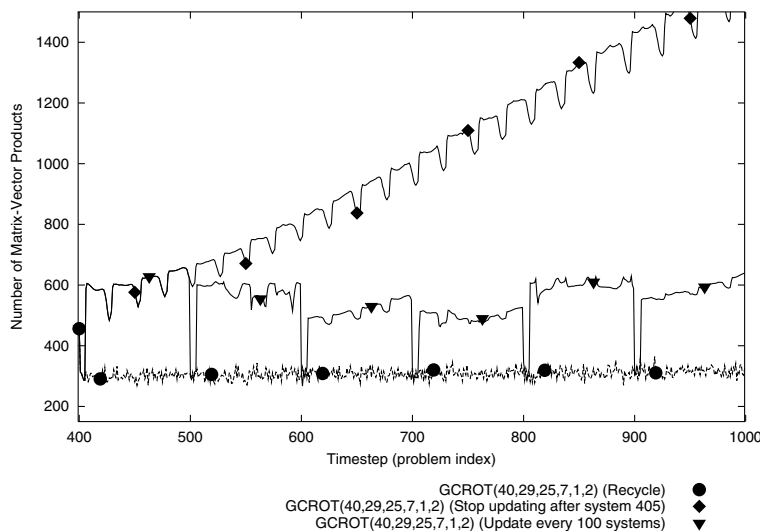


FIG. 4.6. Number of matrix-vector products versus timestep for $GCROT(40, 29, 25, 7, 1, 2)$ with subspace recycling on the nonpreconditioned crack propagation problem. We also show a modified version of $GCROT(40, 29, 25, 7, 1, 2)$ that stops updating the recycled subspace after linear system 405, and a version that stops updating after linear system 405 but runs standard $GCROT$ every 100th system thereafter. In the latter case, convergence remains small for about four or five systems after using standard $GCROT$, but then quickly deteriorates, showing the importance of updating the approximate invariant subspace regularly. Note that $GCROT$, too, is able to restore rapid convergence in the course of a single linear system.

4.2. Electronic structure. First-principles, electronic-structure calculations based on the Schrödinger equation are used to predict key physical properties of materials systems with a large number of atoms. We consider systems arising in the KKR (Korringa–Kohn–Rostoker) method [16, 17] and seek to compute entries in the inverse of the matrix

$$G = (I - (t - t_{\text{ref}})G_{\text{ref}})^{-1}(t - t_{\text{ref}}),$$

where G_{ref} is a sparse and easily invertible matrix and t and t_{ref} are block-diagonal matrices. G is a sparse, complex, non-Hermitian matrix whose relative number of nonzeros decreases with the number of atoms [14, 40, 32].

Only the block-diagonal elements of G^{-1} are needed to calculate physical properties, such as charge densities, total energy, force, and formation and defect energies. As such, we solve $GX = I$, column-by-column. Iterative methods offer the advantage of storing only those components of the inverse that we need. Standard direct inversion methods are infeasible for large numbers of atoms ($N \geq 500$) on regular workstations, because the memory and computational costs grow as $O(N^2)$ and $O(N^3)$, respectively.

We consider a small model problem provided by Duane Johnson (Materials Science and Engineering, UIUC) and Andrei Smirnov (Oak Ridge National Laboratory). The problem involves the simulation of a cubic lattice of 54 copper atoms (treated as inequivalent) for a complex energy point close to the real axis. This is the key physical regime for metals and leads to problems that converge poorly. The matrix is 864×864 and has about 300,000 nonzeros. However, for increasingly larger systems the matrix

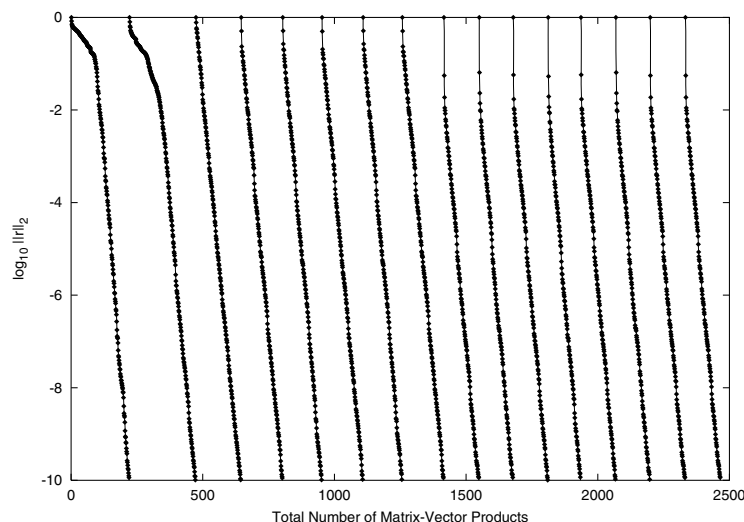


FIG. 4.7. Convergence for 16 consecutive right-hand sides for a small electronic structure problem. Each distinct curve gives the convergence for a subsequent right-hand side, plotted against the total number of matrix-vector products. The first two right-hand sides together take about 500 iterations, while the remaining right-hand sides take about 140 iterations each, a reduction of almost 50%.

becomes more sparse; the number of nonzeros grows roughly linearly with the size of the matrix. We solved this problem using GCRO-DR(50,25) with subspace recycling for 32 consecutive right-hand sides (the first 32 unit Cartesian basis vectors). We give the convergence history for the first atom in Figure 4.7. Note that the first two right-hand sides together take about 500 iterations; the remaining right-hand sides take approximately 140 iterations each, a reduction of almost 50%. Each right-hand side for the second atom (not shown) also takes approximately 140 iterations. Although for problems of this size iterative methods are not competitive with direct solvers, we have observed this convergence behavior for larger problems, in particular the immediate acceleration in convergence for subsequent right-hand sides.

4.3. QCD. Quantum chromodynamics (QCD) is the fundamental theory describing the strong interaction between quarks and gluons. Numerical simulations of QCD on a four-dimensional space-time lattice are considered the only way to solve QCD ab initio [4, 35]. As the problem has a 12×12 block structure, we are often interested in solving for 12 right-hand sides related to a single lattice site. The linear system to be solved is $(I - \kappa D)x = b$ with $0 \leq \kappa < \kappa_c$, where D is a sparse, complex, non-Hermitian matrix representing periodic nearest neighbor coupling on the four-dimensional space-time lattice [19]. For $\kappa = \kappa_c$ the system becomes singular. The physically interesting case is for κ slightly smaller than κ_c ; κ_c depends on D .

As a model problem we use the matrix `conf5.0_0014x4.1000.mtx` downloaded from the Matrix Market website at NIST [2]. The model problems were submitted by Medeke [19]. For this problem we have $\kappa_c = 0.20611$, and we used $\kappa = 0.202$.

We solve for 12 consecutive right-hand sides (the first 12 Cartesian basis vectors) using the GCROT method with subspace recycling. The results are presented in Figure 4.8.

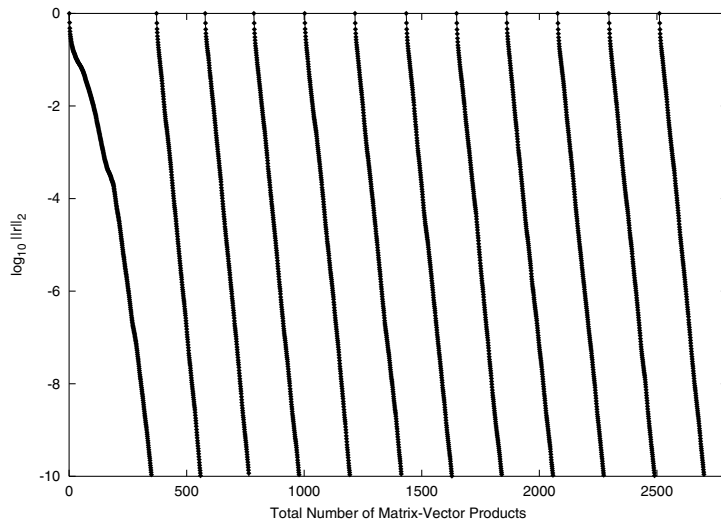


FIG. 4.8. Convergence for 12 consecutive right-hand sides for a model QCD problem from the NIST Matrix Market. Each distinct curve gives the convergence for a subsequent right-hand side, plotted against the total number of matrix-vector products. Note that the final convergence rate for the first system is the starting convergence rate for the rest of the systems.

4.4. Convection-diffusion. We consider the finite difference discretization of the partial differential equation

$$u_{xx} + u_{yy} + cu_x = 0$$

on $(0, 1) \times (0, 1)$ with boundary conditions

$$\begin{aligned} u(x, 0) &= u(0, y) = 0, \\ u(x, 1) &= u(1, y) = 1. \end{aligned}$$

Central differences are used, and we set the mesh width to be $h = 1/41$ in both directions, which results in a 1600×1600 matrix. We consider the symmetric case, $c = 0$, and a nonsymmetric case for $c = 40$. The eigenvalues are real for both these examples. In order to study how a recycled subspace affects convergence, we will consider the “ideal” situation for subspace recycling by solving a linear system *twice* with GCRO-DR and GCROT, recycling the subspace generated from the first run.

In this example, we consider GMRES(∞), GMRES(25), GMRES-DR(25,10), GCRO-DR(25,10), and GCROT(25,18,15,5,1,1). To explore the effects of subspace recycling on this example problem, we *rerun* GCRO-DR and GCROT on the same linear system, and recycle the subspace from the first run. We do this to exclude the effects of right-hand sides having slightly different eigenvector decompositions. In a sense, this is the ideal case for subspace recycling. The first run for GCRO-DR is the same as GMRES-DR. The results for the $c = 40$ (nonsymmetric) case are quite interesting and counterintuitive. In particular, this example suggests that there are better choices for a recycle space than approximate invariant subspaces. This topic is discussed further in [23]. We give the results for the symmetric case, $c = 0$, in Figure 4.9 and for the nonsymmetric case with $c = 40$ in Figure 4.10. In the legend for each of these figures, *Recycle* denotes the second run of a solver that was run twice. All solvers were required to reduce the relative residual to 1.0×10^{-10} .

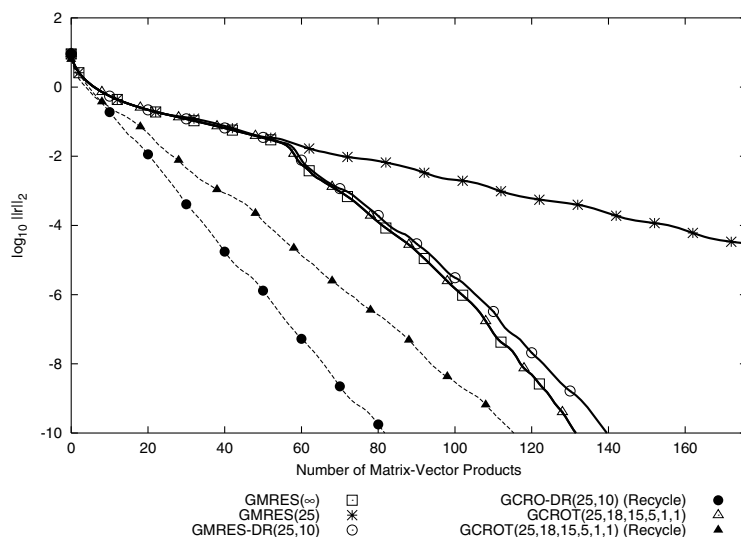


FIG. 4.9. Residual norm versus number of matrix-vector products for various solvers for the convection-diffusion problem with $c = 0$.

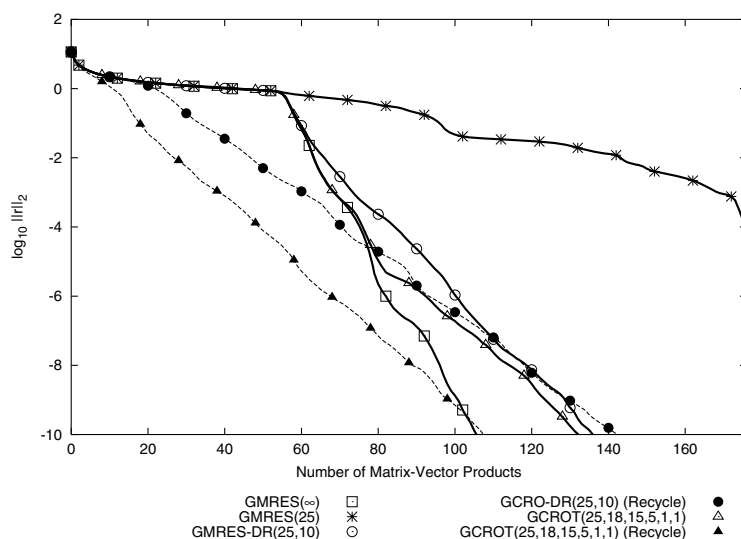


FIG. 4.10. Residual norm versus number of matrix-vector products for various solvers for the convection-diffusion problem with $c = 40$.

For the $c = 0$ case, we see that the second runs of GCRO-DR and GCROT both converged faster than GMRES(∞). Note that on its first run GCROT converges as fast as full GMRES and slightly faster than GCRO-DR/GMRES-DR. On the rerun, however, GCRO-DR is faster than GCROT, suggesting a more effective recycle space. Clearly, GCRO-DR and GCROT recycled a small subspace from their first run that improved convergence significantly. For the $c = 40$ case, GMRES(∞) and the second run of GCROT terminate in about the same number of iterations, but the second run of GCROT had a significantly smaller residual for almost the entire run. Only near

TABLE 4.3

Cosines of principal angles between the recycled subspace and the invariant subspaces spanned by the 10 and 21 eigenvectors associated with the eigenvalues of smallest magnitude, respectively, for the $c = 0$ and $c = 40$ cases.

Cosines of principal angles between recycled subspace and subspace associated with 10 smallest magnitude eigenvalues		Cosines of principal angles between recycled subspace and subspace associated with 21 smallest magnitude eigenvalues	
$c = 0$	$c = 40$	$c = 0$	$c = 40$
1.000000000000000	1.000000000000000	1.000000000000000	1.000000000000000
1.000000000000000	0.999999999999997	1.000000000000000	1.000000000000000
1.000000000000000	0.99999999839942	1.000000000000000	1.000000000000000
1.000000000000000	0.99999970490203	1.000000000000000	0.999999999999937
0.99999999999703	0.99990149788562	1.000000000000000	0.99999999545394
0.00000000593309	0.98844658524616	1.000000000000000	0.99999681064565
0.00000000003840	0.89957454665058	0.999999999999988	0.99983896006215
0.00000000000003	0.54237185670110	0.99999999316379	0.99393007943547
0.00000000000000	0.06426938073642	0.99993817690380	0.94584519976471
0.00000000000000	0.02603228754605	0.99792215267787	0.20867650942988

the end, with a much larger search space, does GMRES(∞) catch up. The second run of GCROT also does better than its first run, indicating that it recycled a subspace useful for convergence. However, GCRO-DR performed initially somewhat better on the second run than the first, but the overall iteration count was approximately the same for both runs. This means that the subspace it recycled failed to improve convergence. For more analysis on the selection of good recycle spaces, see [23, 15, 24].

Table 4.3 shows the cosines of the principal angles between the subspace recycled by GCRO-DR and the invariant subspaces associated with the 10 and 21 eigenvalues of smallest magnitude, respectively, for the $c = 0$ and $c = 40$ cases. For the comparison with 10 eigenvectors, we see that the recycle space for the $c = 0$ case captures only 5 eigenvectors. We compare with the space spanned by 21 eigenvectors because it captures the entire recycled subspace for the $c = 0$ case. This means that GCRO-DR does not select the invariant subspace spanned by the eigenvectors for the 10 smallest eigenvalues, but rather selects some subspace of the space spanned by the 21 smallest. The table also shows that the approximation of an invariant subspace for the $c = 40$ case is nearly as good as for $c = 0$. However, this does not lead to similar convergence.

5. Conclusions and future work. We have presented an overview of Krylov subspace recycling for sequences of linear systems, where both the matrix and right-hand side change. Different choices for subspace selection and recycling have been shown, as well as methods implementing those choices. We propose the new solver GCRO-DR to implement Krylov subspace recycling of approximate invariant subspaces for Hermitian and non-Hermitian systems. We provide two important theoretical results complemented by a set of experiments to analyze the convergence of GCRO-DR for a typical application generating a long sequence of linear systems. GCROT has convergence behavior similar to GCRO-DR regarding recycling, but theoretical results for this method are a topic of future research. When solving a sequence of linear systems, methods employing Krylov subspace recycling frequently outperformed GMRES(∞) while recycling a subspace of only small dimension and minimizing over a small subspace. However, as particular examples in section 4.4 show, it is not completely clear how subspace selection affects convergence, so further theory is needed. Short recurrence methods for the Hermitian case have been developed [15].

Appendix. GCRO with deflated restarting (GCRO-DR).

- 1: Choose m , the maximum size of the subspace, and k , the desired number of approximate eigenvectors. Let tol be the convergence tolerance. Choose an initial guess x_0 . Compute $r_0 = b - Ax_0$, and set $i = 1$.
- 2: **if** \tilde{Y}_k is defined (from solving a previous linear system) **then**
- 3: Let $[Q, R]$ be the reduced QR-factorization of $A\tilde{Y}_k$.
- 4: $C_k = Q$
- 5: $U_k = \tilde{Y}_k R^{-1}$
- 6: $x_1 = x_0 + U_k C_k^H r_0$
- 7: $r_1 = r_0 - C_k C_k^H r_0$
- 8: **else**
- 9: $v_1 = r_0 / \|r_0\|_2$
- 10: $c = \|r_0\|_2 e_1$
- 11: Perform m steps of GMRES, solving $\min \|c - \underline{H}_m y\|_2$ for y and generating V_{m+1} and \underline{H}_m .
- 12: $x_1 = x_0 + V_m y$
- 13: $r_1 = V_{m+1}(c - \underline{H}_m y)$
- 14: Compute the k eigenvectors \tilde{z}_j of $(H_m + h_{m+1,m}^2 H_m^{-H} e_m e_m^H) \tilde{z}_j = \tilde{\theta}_j \tilde{z}_j$ associated with the smallest magnitude eigenvalues $\tilde{\theta}_j$ and store in P_k .
- 15: $\tilde{Y}_k = V_m P_k$
- 16: Let $[Q, R]$ be the reduced QR-factorization of $\underline{H}_m P_k$.
- 17: $C_k = V_{m+1} Q$
- 18: $U_k = \tilde{Y}_k R^{-1}$
- 19: **end if**
- 20: **while** $\|r_i\|_2 > tol$ **do**
- 21: $i = i + 1$
- 22: Perform $m - k$ Arnoldi steps with the linear operator $(I - C_k C_k^H)A$, letting $v_1 = r_{i-1} / \|r_{i-1}\|_2$ and generating V_{m-k+1} , \underline{H}_{m-k} , and B_{m-k} .
- 23: Let D_k be a diagonal scaling matrix such that $\tilde{U}_k = U_k D_k$, where the columns of \tilde{U}_k have unit norm.
- 24: $\hat{V}_m = [\tilde{U}_k \quad V_{m-k}]$
- 25: $\hat{W}_{m+1} = [C_k \quad V_{m-k+1}]$
- 26: $\underline{G}_m = \begin{bmatrix} D_k & B_{m-k} \\ 0 & \underline{H}_{m-k} \end{bmatrix}$
- 27: Solve $\min \|\hat{W}_{m+1}^H r_{i-1} - \underline{G}_m y\|_2$ for y .
- 28: $x_i = x_{i-1} + \hat{V}_m y$
- 29: $r_i = r_{i-1} - \hat{W}_{m+1} \underline{G}_m y$
- 30: Compute the k eigenvectors \tilde{z}_i of $\underline{G}_m^H \underline{G}_m \tilde{z}_i = \tilde{\theta}_i \underline{G}_m^H \hat{W}_{m+1}^H \hat{V}_m \tilde{z}_i$ associated with smallest magnitude eigenvalues $\tilde{\theta}_i$ and store in P_k .
- 31: $\tilde{Y}_k = \hat{V}_m P_k$
- 32: Let $[Q, R]$ be the reduced QR-factorization of $\underline{G}_m P_k$.
- 33: $C_k = \hat{W}_{m+1} Q$
- 34: $U_k = \tilde{Y}_k R^{-1}$
- 35: **end while**
- 36: Let $\tilde{Y}_k = U_k$ (for the next system).

Acknowledgments. We thank one of the anonymous referees and the editor for their helpful comments and for making some very useful suggestions regarding the experimental analysis.

REFERENCES

- [1] C. BEATTIE, M. EMBREE, AND J. ROSSI, *Convergence of restarted Krylov subspaces to invariant subspaces*, SIAM J. Matrix Anal. Appl., 25 (2004), pp. 1074–1109.
- [2] R. F. BOISVERT, R. POZO, K. REMINGTON, R. F. BARRETT, AND J. J. DONGARRA, *Matrix market: A Web resource for test matrix collections*, in Quality of Numerical Software: Assessment and Enhancement, Ronald F. Boisvert, ed., Chapman and Hall, London, 1997, pp. 125–136.
- [3] T. F. CHAN AND M. K. NG, *Galerkin projection methods for solving multiple linear systems*, SIAM J. Sci. Comput., 21 (1999), pp. 836–850.
- [4] M. CREUTZ, *Quarks, Gluons, and Lattices*, Cambridge University Press, Cambridge, UK, 1986.
- [5] E. DE STURLER, *Inner-outer methods with deflation for linear systems with multiple right-hand sides*, in Householder Symposium XIII, Proceedings of the Householder Symposium on Numerical Algebra, 1996, pp. 193–196.
- [6] E. DE STURLER, *Nested Krylov methods based on GCR*, J. Comput. Appl. Math., 67 (1996), pp. 15–41.
- [7] E. DE STURLER, *Truncation strategies for optimal Krylov subspace methods*, SIAM J. Numer. Anal., 36 (1999), pp. 864–889.
- [8] J. J. DONGARRA, I. S. DUFF, D. C. SORESENSEN, AND H. A. VAN DER VORST, *Numerical Linear Algebra for High-Performance Computers*, Software Environ. Tools 7, SIAM, Philadelphia, PA, 1998.
- [9] M. EIERMANN, O. G. ERNST, AND O. SCHNEIDER, *Analysis of acceleration strategies for restarted minimal residual methods*, J. Comput. Appl. Math., 123 (2000), pp. 261–292.
- [10] C. FARHAT AND F.-X. ROUX, *Implicit parallel processing in structural mechanics*, in Computational Mechanics Advances, J. Tinsley Oden, ed., North-Holland, Amsterdam, 1994, Vol. 2, pp. 1–124.
- [11] P. F. FISCHER, *Projection techniques for iterative solution of $A\mathbf{x} = \mathbf{b}$ with successive right-hand sides*, Comput. Methods Appl. Mech. Engrg., 163 (1998), pp. 193–204.
- [12] G. H. GOLUB AND C. F. VAN LOAN, *Matrix Computations*, 3rd ed., Johns Hopkins University Press, Baltimore, MD, 1996.
- [13] A. GULLERUD AND R. H. DODDS, *MPI-based implementation of a PCG solver using an EBE architecture and preconditioner for implicit, 3-D finite element analyses*, Comput. & Structures, 79 (2001), pp. 553–575.
- [14] D. D. JOHNSON, D. M. NICHOLSON, F. J. PINSKI, B. L. GYORFFY, AND G. M. STOCKS, *Energy and pressure calculations for random substitutional alloys*, Phys. Rev. B, 41 (1990), pp. 9701–9716.
- [15] M. E. KILMER AND E. DE STURLER, *Recycling subspace information for diffuse optical tomography*, SIAM J. Sci. Comput., 27 (2006), pp. 2140–2166.
- [16] W. KOHN AND N. ROSTOKER, *Solution of the Schrödinger equation in periodic lattices with an application to metallic lithium*, Phys. Rev., 94 (1954), pp. 1111–1120.
- [17] J. KORRINGA, *On the calculation of the energy of a Bloch wave in a metal*, Phys., 8 (1947), pp. 392–400.
- [18] G. MACKAY, *Reusing Krylov Subspaces for Sequences of Linear Systems*, Master's thesis, Department of Computer Science, University of Illinois at Urbana-Champaign, 2003.
- [19] B. MEDEKE, *Set QCD: Quantum Chromodynamics*, description of matrix set on NIST Matrix Market, <http://math.nist.gov/MatrixMarket>.
- [20] R. B. MORGAN, *Implicitly restarted GMRES and Arnoldi methods for nonsymmetric systems of equations*, SIAM J. Matrix Anal. Appl., 21 (2000), pp. 1112–1135.
- [21] R. B. MORGAN, *GMRES with deflated restarting*, SIAM J. Sci. Comput., 24 (2002), pp. 20–37.
- [22] D. P. O'LEARY, *The block conjugate gradient algorithm and related methods*, Linear Algebra Appl., 29 (1980), pp. 293–322.
- [23] M. L. PARKS, *The Iterative Solution of a Sequence of Linear Systems Arising from Nonlinear Finite Element Analysis*, Ph.D. thesis, Department of Computer Science, University of Illinois at Urbana-Champaign, 2005.
- [24] M. L. PARKS AND E. DE STURLER, *Analysis of Krylov subspace recycling for sequences of linear systems*, 2006, to appear.
- [25] C. REY AND F. RISLER, *A Rayleigh-Ritz preconditioner for the iterative solution to large-scale nonlinear problems*, Numer. Algorithms, 17 (1998), pp. 279–311.

- [26] F. RISLER AND C. REY, *Iterative accelerating algorithms with Krylov subspaces for the solution to large-scale nonlinear problems*, Numer. Algorithms, 23 (2000), pp. 1–30.
- [27] Y. SAAD, *Iterative Methods for Sparse Linear Systems*, 2nd ed., SIAM, Philadelphia, PA, 2003.
- [28] Y. SAAD AND M. H. SCHULTZ, *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 856–869.
- [29] Y. SAAD, M. YEUNG, J. ERHEL, AND F. GUYOMARC'H, *A deflated version of the conjugate gradient algorithm*, SIAM J. Sci. Comput., 21 (2000), pp. 1909–1926.
- [30] V. SIMONCINI AND E. GALLOPOULOS, *An iterative method for nonsymmetric systems with multiple right-hand sides*, SIAM J. Sci. Comput., 16 (1995), pp. 917–933.
- [31] V. SIMONCINI AND D. B. SZYLD, *On the occurrence of superlinear convergence of exact and inexact Krylov subspace methods*, SIAM Rev., 47 (2005), pp. 247–272.
- [32] A. V. SMIRNOV AND D. D. JOHNSON, *Accuracy and limitations of localized Green's function methods for materials science applications*, Phys. Rev. B, 64 (2001), paper 235129.
- [33] A. STATHOPOULOS, Y. SAAD, AND K. WU, *Dynamic thick restarting of the Davidson, and the implicitly restarted Arnoldi methods*, SIAM J. Sci. Comput., 19 (1998), pp. 227–245.
- [34] G. W. STEWART, *Matrix Algorithms. Volume II: Eigensystems*, SIAM, Philadelphia, PA, 2001.
- [35] J. VAN DEN ESHOF, A. FROMMER, TH. LIPPERT, K. SCHILLING, AND H. A. VAN DER VORST, *Numerical methods for the QCD overlap operator: I Sign function and error bounds*, Comput. Phys. Comm., 146 (2002), pp. 203–224.
- [36] H. A. VAN DER VORST AND C. VUIK, *The superlinear convergence behaviour of GMRES*, J. Comput. Appl. Math., 48 (1993), pp. 327–341.
- [37] B. VITAL, *Etude de quelques méthodes de résolution de problèmes linéaires de grande taille sur multiprocesseur*, Ph.D. thesis, Université de Rennes I, Rennes, France, 1990.
- [38] K. WU AND H. SIMON, *Thick-restart Lanczos method for large symmetric eigenvalue problems*, SIAM J. Matrix Anal. Appl., 22 (2000), pp. 602–616.
- [39] U. MEIER YANG AND K. A. GALLIVAN, *A new family of block methods*, Appl. Numer. Math., 30 (1999), pp. 155–173.
- [40] R. ZELLER, P. H. DEDERICHS, B. UJFALUSSY, L. SZUNYOGH, AND P. WEINBERGER, *Theory and convergence properties of the screened Korringa-Kohn-Rostoker method*, Phys. Rev. B, 52 (1995), pp. 8807–8812.