RECYCLING KRYLOV SUBSPACES FOR SEQUENCES OF LINEAR SYSTEMS*

MICHAEL L. PARKS†, ERIC DE STURLER†, GREG MACKEY†, DUANE D. JOHNSON‡, AND SPANDAN MAITI§

Technical Report UIUCDCS-R-2004-2421 (CS), UILU-ENG-2004-1722 (ENGR)
March 2004

Abstract. Many problems in engineering and physics require the solution of a large sequence of linear systems. We can reduce the cost of solving subsequent systems in the sequence by recycling information from previous systems. We overview two different approaches. For several model problems, we demonstrate that we can reduce the iteration count required to solve a linear system by a factor of two. We consider both Hermitian and non-Hermitian problems, and present numerical experiments to illustrate the effects of subspace recycling.

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

AMS subject classifications. 65F10

1. Introduction.

We consider the solution of a sequence of general linear systems

$$A^{(i)}x^{(i)} = b^{(i)}, \qquad i = 1, 2, \ldots, \tag{1.1}$$

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 are typically not available simultaneously. Such sequences arise in many problems, such as Newton or Broyden-type methods for solving nonlinear equations. They also occur in modeling fatigue and fracture via finite element analysis. These analyses use dynamic loading, requiring many loading steps, and rely on implicit solvers [14]. Generally, several thousand loading increments 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. We are interested in retaining a subspace determined while solving previous systems and use it to reduce the cost of solving the next system. We refer to this process as *Krylov subspace recycling.*

For the Hermitian positive definite case, Rey and Risler have proposed to reduce the effective condition number by retaining all converged Ritz vectors arising in a previous CG iteration [24, 25, 26]. In general, this requires significant storage. Moreover, memory-wise, they lose the advantage of a short recurrence, as they keep the full recurrence during the solution of a single system. Since they focus on the finite

[*] The work of M. Parks was supported by the CSE program at UIUC through two CSE fellowships. The work of E. de Sturler was supported by the Materials Computation Center through grant NSF DMR 99-76550 and the Center for Simulation of Advanced Rockets through grant DOE LLNL B341494. The work of G. Mackey was supported by the Center for Simulation of Advanced Rockets through grant DOE LLNL B341494. The work of D. D. Johnson was supported through the Materials Computation Center through grant NSF DMR 99-76550.

[†] Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, IL 61801 (parks@uiuc.edu, sturler@cs.uiuc.edu, gmackey@uiuc.edu).

[‡] Department of Materials Science, University of Illinois at Urbana-Champaign, Urbana, IL 61801 (duanej@uiuc.edu).

[§] Department of Aeronautical and Astronautical Engineering, University of Illinois at Urbana-Champaign, Urbana, IL 61801.

element tearing and interconnecting (FETI) method [11], it is less of a drawback, because the interface problem is small relative to the overall problem, and it is common to use a full recurrence in FETI. The two Galerkin projection methods developed by Chan and Ng [3] could also be used. These methods require all systems to be available simultaneously, or at least the right hand sides. Moreover, they focus on situations where all the matrices are very close. However, for the problems we target, the matrices change only slowly, but the incremental change over many steps can be significant.

Solving a sequence of linear systems where the matrix is invariant is a special case of (1.1). When all right hand sides are available simultaneously, block methods such as block CG [23], block GMRES [34], and the family of block EN-like methods [35] are often suitable. 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 [12], the deflated conjugate gradient method (deflated CG) [29], or the hybrid method of Simoncini and Gallopoulos [30] may be employed. Fischer's method first looks for a solution in the space spanned by the previous solution vectors in the sequence, which is only helpful if the solution vectors are correlated. In deflated CG, only a small number of the initial Lanczos vectors for every system are used to update the approximate invariant subspace. This is efficient, both 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 only when the right hand sides share common spectral information.

When solving (1.1), we should consider:

1. Which subspace should be recycled for the next system?
2. How should it be used?

We discuss two answers to the first question. One idea is to recycle an approximate invariant subspace and use it for deflation. Clearly, reducing the effective condition number of a matrix may speed convergence. An alternative idea is to recycle a subspace that minimizes the loss of orthogonality with the Krylov subspace from the previous system [6]. We elaborate on the latter choice in section 2.3.

We discuss three answers to the second question. We refer to these approaches as:

- *augmentation*,
- *orthogonalization*,
- *preconditioning*.

In an augmentation approach, we append additional vectors at the end of the Arnoldi recurrence, in the manner of FGMRES, such that an Arnoldi-like relation is formed [27]. In an orthogonalization approach, we first minimize the residual over the recycled subspace, and then maintain orthogonality with the image of this space in the Arnoldi recurrence. In a preconditioning approach, we construct preconditioners that shift eigenvalues [1, 10]. When using exactly invariant subspaces, an augmentation approach is superior to a preconditioning approach [8]. Hence, we consider only the augmentation and orthogonalization approaches.

In secton 2, we discuss several truncated or restarted linear solvers that use the ideas above to reduce the total number of iterations for solving a sequence of linear systems. We define a *cycle* as the computation between truncations or restarts. Subspaces that are useful to retain for a subsequent cycle when solving a single linear system may also be useful for subsequent linear systems in a sequence, especially if the matrix does not change significantly. Therefore, we consider linear solvers that retain a carefully selected subspace after each cycle. Several such solvers have been proposed. We consider Morgan's GMRES-DR [22] and de Sturler's GCROT [6], and modify GCROT to recycle subspaces between linear systems. GMRES-DR cannot be modified to do this, so we introduce GCRO-DR, a flexible variant of GMRES-DR capable of Krylov subspace recycling.

In section 3, we introduce several test problems, including both realistic problems taken from engineering and physics, as well as a problem constructed explicitly for analysis of subspace recycling. In section 4, we give the experimental results, which show that recycling can be very beneficial. Conclusions and future work are given in section 5.

2. Truncated and Augmented 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 carefully selected subspace between cycles. A taxonomy of popular choices is given in [8]. In this section, we discuss those choices and solvers implementing them. We then investigate how those solvers might be modified to recycle subspaces between linear systems.

Morgan's GMRES-DR and GMRES-E [20] retain an approximately invariant subspace between cycles. In particular, both methods focus on removing the eigenvalues of smallest magnitude, and retain a subspace spanned by approximate eigenvectors associated with those eigenvalues. GMRES-E uses an augmentation approach, which was analyzed in [27]. In contrast, GMRES-DR uses an orthogonalization approach. Despite these differences, GMRES-E and GMRES-DR generate the same Krylov subspace at the end of each cycle if they retain the same harmonic Ritz vectors; see [20, 22]. Although GMRES-E retains the same subspace between cycles as GMRES-DR, GMRES-E can be modified to select any subspace, whereas GMRES-DR cannot. Thus, GMRES-E is suitable for Krylov subspace recycling between systems, as in (1.1). GMRES-DR cannot be modified for Krylov subspace recycling, even when the matrix does not change. We discuss GMRES-E and GMRES-DR further in section 2.4. Because GMRES-DR cannot be used for Krylov subspace recycling, we combine ideas from GCRO [5] and GMRES-DR to produce a new linear solver, GCRO-DR. GCRO-DR 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 2.5. In section 2.6 we give some analysis suggesting why recycling nearly invariant subspaces may improve convergence.

Another strategy for subspace selection was proposed in [6] and was used for the GCROT method, an extension of GCRO. We discuss this approach, and its modification towards solving (1.1) in section 2.3.

We first review some definitions.

2.1. Definitions.

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

$$AV_m = V_{m+1}\overline{H}_m, \tag{2.1}$$

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

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

$$Ay - \theta y \perp w, \qquad \forall w \in S. \tag{2.2}$$

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,

$$A^{-1}\widetilde{y} - \widetilde{\mu}\widetilde{y} \perp w \qquad \forall w \in AS, \tag{2.3}$$

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

2.2. GMRES and GCR.

We now review the linear solvers GMRES [28] and GCR [9], which form the basis for the linear solvers we discuss later. The Arnoldi iteration is the core of GMRES. When solving $Ax = b$ with GMRES, we start with an initial guess $x_0 \in \mathbb{C}^n$ and compute the initial residual $r_0 = b - Ax_0$. Let the first Arnoldi vector be $v_1 = r_0/\|r_0\|_2$. We proceed with m Arnoldi iterations to form relation (2.1) with $range(V_m) = K^m(A, r_0)$. Then, we solve $\min \|c - \overline{H}_m d\|_2$ for $d \in \mathbb{C}^m$, where $c = \|r_0\|_2 e_1$. Finally, we form the new approximate solution, $x_m = x_0 + V_m d$. GMRES solves the least squares problem $A(x_0 + V_m d) \approx r_0$ for d. So, $r_m \perp AK^{(m)}(A, r_0)$.

The linear solver GCR is algebraically equivalent to GMRES, but requires more storage, as it keeps separate bases for $K^{(m)}(A, r_0)$ and $AK^{(m)}(A, r_0)$. GCR maintains the matrices $U_m, C_m \in \mathbb{C}^{n\times m}$, so that

$$range(U_m) = K^{(m)}(A, r_0), \tag{2.4}$$

$$AU_m = C_m, \tag{2.5}$$

$$C_m^H C_m = I_m. \tag{2.6}$$

We solve the minimization problem $\min \|r_0 - AU_m d\|_2$ for $d \in \mathbb{C}^m$, and compute the solution as $x_m = x_0 + U_m d = x_0 + U_m C_m^H r_0$, and residual as $r_m = r_0 - C_m C_m^H r_0 \perp AK^{(m)}(A, r_0)$. The relations (2.5)-(2.6) still hold if $range(U_m)$ is not a Krylov space, allowing us to find the minimum residual solution over any subspace $range(U_m)$. In this case the method would not be called GCR, but the relations (2.5)-(2.6) are still valid.

2.3. 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 truncated space is minimized. This process is called *optimal truncation*.

We discuss the idea of optimal truncation in the context of restarted GMRES, although it can be described in more general terms, and independently of any specific linear solver [6, 18]. Consider solving $Ax = b$ with initial residual r_0. The idea is to determine, after each cycle, a subspace to retain for the next cycle in order to maintain good convergence after the restart. 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 $s < m$. After s iterations of GMRES, we have the Arnoldi relation

$$AV_s = V_{s+1}\overline{H}_s. \tag{2.7}$$

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* [13, pp. 603–4] between the two subspaces $AK^{(s)}(A,r_0)$ and $AK^{(m-s)}(A,r)$. Optimal truncation involves selecting and retaining a k-dimensional subspace of $AK^{(s)}(A,r_0)$ such that the magnitude of the residual error $\|e\|_2 = \|r_1 - r_2\|_2$, is minimized. The complement of that subspace is discarded. 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 and C_k satisfying the relations (2.5)-(2.6). The minimum residual solution over $range(U_k)$ 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, but we take the singularity of the operator into account. Hence, GCROT uses an orthogonality approach. 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 $\mathrm{O}(nm^2 + nkm)$ other floating point operations. For details, see [6].

GCROT can be modified to solve (1.1) by carrying over U_k from the i^{th} system to the $(i+1)^{st}$ system. After we solve the i^{th} system $A^{(i)}x^{(i)} = b^{(i)}$ with GCROT, we have the relation $A^{(i)}U_k = C_k$. We must modify U_k and C_k so that (2.5)-(2.6) hold with respect to $A^{(i+1)}$, which we do as follows:

```
1: $[Q,R]$ = reduced QR decomposition of $A^{(i+1)}U_k^{old}$
2: $C_k^{new} = Q$
3: $U_k^{new} = U_k^{old}R^{-1}$
```

Now, $A^{(i+1)}U_k^{new} = C_k^{new}$, and we can proceed with GCROT on the system $A^{(i+1)}x^{(i+1)} = b^{(i+1)}$. 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 3.1. Moreover, even if we were to compute $A^{(i+1)}U_k^{old}$ directly, this can be faster than k separate matrix-vector multiplications [7]. 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.

2.4. GMRES-DR and GMRES-E. GMRES-DR and GMRES-E rely on spectral or nearly invariant subspace information, rather than orthogonality constraints. Removing or deflating certain eigenvalues can greatly improve convergence. Based on this idea, Morgan has proposed three linear solvers (GMRES-E, GMRES-IR [21], and GMRES-DR) that aim to deflate the eigenvalues of smallest magnitude. However, these solvers can be changed to deflate other eigenvalues. We consider only GMRES-E and GMRES-DR.

GMRES-E (GMRES with eigenvectors) appends harmonic Ritz vectors after the Arnoldi recurrence, resulting in the Arnoldi-like relation

$$A[V_{m-k} \quad \widetilde{Y}_k] = V_m \overline{H}_m, \tag{2.8}$$

where $v_1 = r_0/\|r_0\|$, $\widetilde{Y}_k = [\widetilde{y}_1, \widetilde{y}_2, \ldots, \widetilde{y}_k]$ contains the k harmonic Ritz vectors from the previous cycle, and where the last k columns of V_m are formed by orthogonalizing the vectors $A\widetilde{y}_i$, for $i = 1 \ldots k$, against the previous columns of V_m. For the first cycle, the harmonic Ritz vectors can be computed from H_m in (2.1). It can be shown that the augmented subspace

$$span\{r_0, Ar_0, A^2 r_0, \ldots, A^{m-k-1} r_0, \widetilde{y}_1, \widetilde{y}_2, \ldots, \widetilde{y}_k\} \tag{2.9}$$

is itself a Krylov subspace, but with another starting vector [21].

GMRES-DR is algebraically equivalent to GMRES-E at the end of each cycle if both select the same harmonic Ritz vectors. Because (2.9) is a Krylov subspace, it means that the harmonic Ritz vectors can go first, rather than being appended at the end. It was shown in [21] that the subspace

$$span\{\widetilde{y}_1, \widetilde{y}_2, \ldots, \widetilde{y}_k, A\widetilde{y}_i, A^2\widetilde{y}_i, \ldots, A^{m-k}\widetilde{y}_i\} \tag{2.10}$$

is identical to subspace (2.9) for $1 \leq i \leq k$. In one cycle, GMRES-DR first orthogonalizes $\widetilde{Y}_k$, giving $\widetilde{\Upsilon}_k$. Then GMRES-DR carries out the Arnoldi recurrence for $m-k$ iterations while maintaining orthogonality to $\widetilde{\Upsilon}_k$. This gives the Arnoldi-like relation

$$A[\widetilde{\Upsilon}_k \quad V_{m-k}] = [\widetilde{\Upsilon}_k \quad V_{m-k+1}]\overline{H}_m, \tag{2.11}$$

where $\overline{H}_m$ is upper Hessenburg, except for a leading dense $(k+1) \times (k+1)$ submatrix. It updates the solution and residual as in GMRES. It then computes the harmonic Ritz vectors associated with the k smallest harmonic Ritz values using (2.11), and finally restarts with those vectors. Note that each column vector in V_{m-k} is orthogonal to $range(\widetilde{Y}_k)$ in GMRES-DR, but this is not true in GMRES-E.

GMRES-DR cannot be directly used to solve (1.1), even if the matrix is invariant. The harmonic Ritz vectors of A in $\widetilde{Y}_k$ do not form a Krylov subspace for another matrix or even just another starting vector. However, Morgan discusses in [22] a modification to GMRES-DR that can be used for the case of multiple right hand sides. Standard GMRES-DR is run for the first right hand side, and the approximate eigenvectors are retained. For subsequent right hand sides, restarted GMRES is used. Between cycles of restarted GMRES, the minimum residual solution over the space spanned by the approximate eigenvectors is found, and the solution and residual vectors updated accordingly. However, the approximate eigenvectors are never updated. We expect this process may suffer the same difficulties as restarted GMRES, such as poor convergence or stagnation. Additionally, for nonsymmetric problems, setting the residual orthogonal to an invariant subspace does not remove that subspace from the residual, which may result in poor convergence.

Because GMRES-E takes an augmentation approach, it can be used when solving (1.1). After the solution of the i^{th} linear system, we could run GMRES on the $i+1^{st}$ linear system for $m-k$ steps, then append the k approximate eigenvectors from the i^{th} linear system to the Arnoldi basis vectors, and then proceed as normal with GMRES-E. This would form the subspace (2.9) for the matrix $A^{(i+1)}$, which is *not* a Krylov subspace. Note that breakdown can occur if a subspace of $\widetilde{Y}_k$ is contained in the Krylov subspace generated first. We observed this when GMRES-E was run on the example problem in section 3.1. Because GMRES-E extends the search space as restarted GMRES, it may suffer from stagnation. Further, the Krylov subspace generated by GMRES-E ignores the orthogonality to $range(A^{(i+1)}\widetilde{Y}_k)$ and thus considers corrections in $range(\widetilde{Y}_k)$ even though the residual is already orthogonal to $range(A^{(i+1)}\widetilde{Y}_k)$. Although GMRES-E can be used when solving (1.1), because of these problems, we do not consider it further. Based on experiments, we believe that it is preferable to preserve orthogonality to $range(A^{(i+1)}\widetilde{Y}_k)$. The linear solver GCRO-DR, discussed next, accomplishes this.

2.5. GCRO-DR. We introduce a new Krylov method that retains a subspace between restarts. We call this method GCRO-DR because it uses deflated restarting within the framework of GCRO [5]. The method is a generalization of GMRES-DR to solve (1.1). GCRO-DR is more flexible because *any* subspace may be retained for subsequent cycles, and also between 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 to solve sequences of linear systems.

Suppose that we solved the i^{th} system of (1.1) with GCRO-DR. We retain k approximate eigenvectors, $\widetilde{Y}_k = [\widetilde{y}_1, \widetilde{y}_2, \ldots, \widetilde{y}_k]$. GCRO-DR maintains matrices $U_k, C_k \in \mathbb{C}^{n\times k}$ such that

$$A^{(i+1)}U_k = C_k, \tag{2.12}$$

$$C_k^H C_k = I_k, \tag{2.13}$$

where U_k and C_k are determined from $\widetilde{Y}_k$ and $A^{(i+1)}$ as follows.

```
$[Q, R]$ = reduced QR decomposition of $A^{(i+1)}\widetilde{Y}_k$
$C_k = Q$
$U_k = \widetilde{Y}_k R^{-1}$
```

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

$$(I - C_k C_k^H)A^{(i+1)}V_{m-k} = V_{m-k+1}\overline{H}_{m-k}. \tag{2.14}$$

Each of the Arnoldi vectors $V_{m-k+1} = [v_1, v_2, \ldots, v_{m-k+1}]$ is orthogonal to $range(C_k)$. We can rewrite (2.14) as

$$A[U_k \quad V_{m-k}] = [C_k \quad V_{m-k+1}] \begin{bmatrix} I_k & B_k \\ 0 & \overline{H}_{m-k} \end{bmatrix} \tag{2.15}$$

where $B_{m-k} = C_k^H A V_{m-k}$. For numerical reasons, we normalize the column vectors of U_k and replace the identity matrix I_k above with a diagonal matrix D_k, such that $U_k D_k$ has unit columns. We denote the rescaled U_k as $\widetilde{U}_k$. Now, the columns of $[\widetilde{U}_k \;\; V_{m-k}]$ and $[C_k \;\; V_{m-k+1}]$ have unit norm, which ensures that the rightmost matrix in (2.15) is not unnecessarily ill-conditioned. This improves the accuracy of the numerical solution.

We define

$$\widehat{V}_m = [\widetilde{U}_k \;\; V_{m-k}], \quad \widehat{W}_{m+1} = [C_k \;\; V_{m-k+1}], \quad \overline{G}_m = \begin{bmatrix} D_k & B_k \\ 0 & \overline{H}_{m-k} \end{bmatrix},$$

and write (2.15) more compactly, as

$$A\widehat{V}_m = \widehat{W}_{m+1}\overline{G}_m. \tag{2.16}$$

Note that $\overline{G}_m = \widehat{W}_{m+1}^H A\widehat{V}_m$ is upper Hessenberg, with D diagonal. The columns of $\widehat{W}_{m+1}$ are orthogonal, but this is not true for the columns of $\widehat{V}_m$.

At each cycle, GCRO-DR solves the minimization problem

$$t = \arg \min_{z \in \ range(\widehat{V}_m)} \|r - Az\|_2, \tag{2.17}$$

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

$$\overline{G}_m y \cong \widehat{W}_{m+1}^H r = \|r\|_2 e_{k+1}, \tag{2.18}$$

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

$$r = r - A\widehat{V}_m y = r - \widehat{W}_{m+1}\overline{G}_m y, \tag{2.19}$$

$$x = x + \widehat{V}_m y. \tag{2.20}$$

To compute new harmonic Ritz vectors the method solves the generalized eigenvalue problem

$$\overline{G}_m^H \overline{G}_m \widetilde{z}_i = \widetilde{\theta}_i \overline{G}_m^H \widehat{W}_{m+1}^H \widehat{V}_m \widetilde{z}_i, \tag{2.21}$$

derived from (2.3), and recovers the harmonic Ritz vectors as $\widetilde{y}_i = \widehat{V}_m \widetilde{z}_i$.

Computationally, GCRO-DR and GMRES-DR use the same number of matrix-vector products per cycle, although a matrix-vector product for GCRO-DR is slightly more expensive, as a modified operator is used. If f is the average number of nonzeros per row in $A^{(i)}$, then the cost of a matrix-vector product for GMRES-DR is $2fn$, and $2fn{+}4kn$ for GCRO-DR, where $k \ll n$. The additional $4kn$ is the cost orthogonalizing against C_k. Both GCRO-DR and GMRES-DR solve a small $m{\times}m$ eigenvalue problem each cycle. GMRES-DR orthonormalizes $k+1$ vectors of length $m+1$ while GCRO-DR finds the QR-factorization of a small $(m+1) \times m$ matrix. Finally, GMRES-DR stores k fewer vectors.

2.6. Recycling Invariant Subspaces. When recycling nearly invariant subspaces, we show a residual bound demonstrating improved convergence under certain assumptions. The following theorem is adapted to our purpose from [31], which was in turn inspired by [27].

THEOREM 2.1. *Let range*(Q_k) *be a* k*-dimensional invariant subspace of* $A \in \mathbb{C}^{n\times n}$. *Let* P_Q *be the spectral projector onto range*(Q_k). *Let range*(Y_k) *be a* k*-dimensional subspace close to range*(Q_k). *Let* P_{AY} *be the orthogonal projector onto*

$range(AY_k)$. Let $range(X_j)$ be a j-dimensional Krylov subspace. Let $r_0 \in \mathbb{C}^n$, $d^* = \arg\min_{d \in range(AY_k)} \|r_0 - d\|_2$, and $r_1 = r_0 - d^*$. Then,

$$\min_{d_1 \in range([AX_j, AY_k])} \|r_0 - d_1\|_2 \leq$$
$$\min_{d_2 \in range(AX_j)} \{\|(I - P_Q)(r_1 - d_2)\|_2 + \gamma \|P_Q(r_1 - d_2)\|_2\},$$

where $\gamma = \|(I - P_{AY})P_Q\|_2$.

Proof. See Theorem 2.1 in [31]. □

Theorem 2.1 in [31] is used to explain superlinear convergence in GMRES as follows. If the Krylov subspace generated by GMRES contains a nearly invariant subspace of $range(A)$, then the GMRES iteration acts nearly as if the residual vector has no components in the associated invariant subspace, resulting in an increased rate of convergence. Our use of the proof is similar, except that we begin our iteration by optimizing over a nearly invariant subspace (recycled from the previous linear system). In the context of the methods we have discussed, we can consider $range(Q_k)$ to span an invariant subspace of A close to the selected k-dimensional approximate invariant subspace, and Y_k to represent the k approximate eigenvectors recycled from the previous system. In the ideal case, $range(Q_k) = range(AY_k)$. The term $\|(I - P_Q)(r_1 - d_2)\|_2$ represents the residual norm achieved by j steps of a Krylov method if the residual r_1 had no components in $range(Q_k)$, and the term $\gamma\|P_Q(r_1 - d_2)\|_2$ will be small if $range(Q_k)$ is sufficiently close to $range(AY_k)$ [31]. We observe that for GCRO-DR, the Krylov subspace $range(X_j)$ is not $X_j = K^{(j)}(A, r_0)$, but is instead $X_j = K^{(j)}((I - P_{AY})A, (I - P_{AY})r_0)$.

3. Test Problems.

We discuss our main example in section 3.1, a problem from fracture mechanics that produces a large 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. As these systems are SPD, we also provide results for three problems that involve real nonsymmetric matrices and complex non-Hermitian matrices. To illustrate the effectiveness of our approach for the case of an invariant matrix, we consider two examples from physics. We discuss electronic structure calculations in section 3.2, and a problem from lattice QCD in section 3.3. Finally, in section 3.4, we apply the two main approaches we have discussed to a simple convection diffusion problem. We use this example to explore the effects of subspace recycling in the nonsymmetric case, independently from perturbations in the matrix or right hand side. We show all methods for the main example, but for brevity we show only selected methods for the remaining problems. Computational results are presented in section 4.

3.1. 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 [14].

We study a sequence of linear systems taken from a finite element code developed by Philippe Geubelle and Spandan Maiti (both Aeronautical and Astronautical Engineering, UIUC). The code simulates crack propagation in a metal plate using so-called “cohesive finite elements”. The plate mesh is shown in Figure 3.1. The problem is

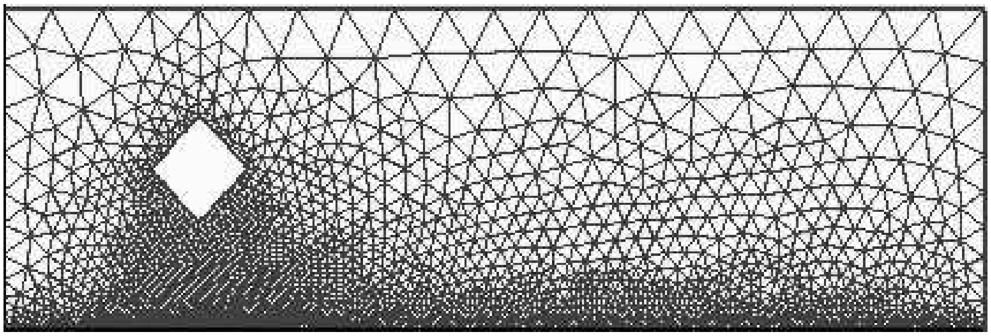


FIG. 3.1. *2D plate mesh for crack propagation problem.*

symmetric about the x-axis, and in this problem 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. These elements are usually inserted dynamically, although that is not the case here. The element stiffness is set to zero for a broken cohesive element. This 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 the 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. They have an average of 13.4 nonzero entries per row. We will consider a sequence of 150 linear systems, both preconditioned and nonpreconditioned. We give results in section 4.1.

3.2. 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 method [17, 16].

For an electron that is not scattered going from atom i to atom j, the Green's function solution is the structural Green's function

$$G_0(r_i, r_j; E) = \frac{e^{i\sqrt{E}|r_i - r_j|}}{4\pi|r_i - r_j|},$$

where r_i and r_j are position vectors, and E is the complex energy. For an electron scattered going from atom i to atom j, the Green's function can be given as follows.

$$G^{ij} = t^i + t^i G_0^{ij} t^j + t^i G_0^{ik} t^k G_0^{kj} t^j + ..., \tag{3.1}$$

where each t^i is a single-site scattering matrix depending only on the local potential. In matrix notation, this recursive definition gives the following equation for G,

$$\begin{aligned} G = t + tG_0(t + tG_0 t + \ldots) = t + tG_0 G \Leftrightarrow \\ (I - tG_0)G = t, \end{aligned} \tag{3.2}$$

where t is the block-diagonal matrix with blocks t^i. A localization strategy transforms (3.2) into an equation for the Green's function relative to a fictitious reference system chosen to ensure localization. This yields a sparse matrix to invert.

$$\begin{aligned} G_{\text{ref}} &= (I - t_{\text{ref}} G_0)^{-1} t_{\text{ref}}, \\ G &= (I - (t - t_{\text{ref}}) G_{\text{ref}})^{-1} (t - t_{\text{ref}}). \end{aligned}$$

The first system above can be inverted very rapidly. The second requires the inversion of a sparse, complex, non-Hermitian matrix, where the relative number of nonzeros in the matrix decreases with the number of atoms [15, 36, 32]. We give results in Section 4.2, using a model problem provided by Duane Johnson (Materials Science, UIUC) and Andrei Smirnov (Oak Ridge National Laboratory).

Only the block-diagonal elements (corresponding to local sites) are needed to calculate physical properties. Iterative methods offer the advantage to store only those components of the inverse (computed column-by-column) 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 $O(N^3)$. Once the electronic Green's function is determined, one can determine important physical properties such as charge densities, total energy, force, formation and defect energies in terms of the Green's function.

3.3. 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, 33]. 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 close to κ_c; κ_c depends on D. We present results in Section 4.3.

3.4. 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 **c=0** case and the nonsymmetric **c=40** case. 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. We show results in section 4.4.

4. Numerical Results.

We explore the effects of subspace recycling by comparing the performance of GCRO-DR and GCROT utilizing subspace recycling with CG, GMRES, restarted GMRES, GMRES-DR, and GCROT without subspace recycling. All of the examples in this section use a zero initial guess. In particular, for the fracture mechanics problem, we solve for the incremental displacement associated

Table 4.1

The total number of iterations required to solve 150 sequential IC(0) preconditioned linear systems is compared. 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**

with the loading increment. In this case, using the previous solution as the initial guess for the next system has no benefit, as the displacements are not correlated. Both preconditioned and nonpreconditioned examples are given.

In the following sections, GMRES(m) indicates restarted GMRES with a maximum subspace of dimension m, and GMRES indicates full (not restarted) GMRES. CG refers to the conjugate gradient method. For GMRES-DR(m, k) and GCRO-DR(m, k), m is the maximum subspace size, and k is the number of vectors kept between cycles. For 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 2.3) is k_{max}. The argument k_{min} indicates the number of column vectors retained in U_k and C_k after truncation. 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 [6, 18] for more regarding the choice of parameters. At each restart, GMRES 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. In this example, we solve a sequence of 150 symmetric positive definite linear systems. Results for nonpreconditioned systems and preconditioned systems are given. Each matrix has a condition number of approximately 10^4, before preconditioning. All solvers were required to reduce the relative residual to 1.0e−10. The number of matrix-vector multiplications required to solve each of these systems is shown in Figure 4.1 for full 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 4.1 minimize over a subspace of dimension 40. GMRES(40) is not shown in Figure 4.1 because it required an order of magnitude more matrix-vector multiplications than the other methods to converge. The results in Figure 4.2 are for the same sequence with an incomplete Cholesky (IC(0)) preconditioner applied to each problem. A new preconditioner was computed for each matrix, which is not the most efficient approach. The number of matrix-vector products to solve all 150 preconditioned linear systems is given in Table 4.1.

We see in Figure 4.1 that GCRO-DR, which employs subspace recycling, requires the fewest matrix-vector products, except for the first system in the sequence, for

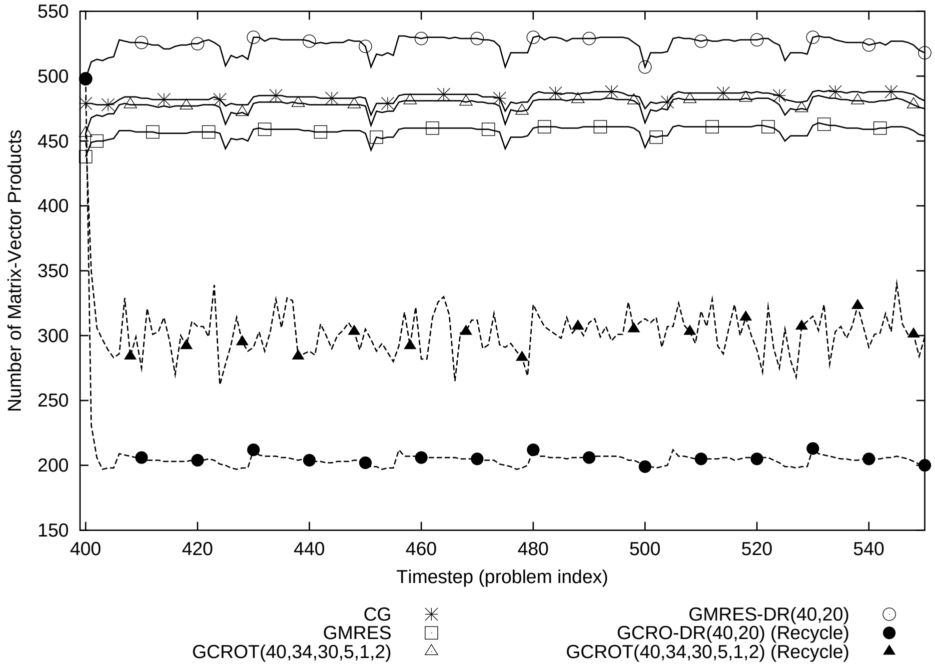


FIG. 4.1. *Number of matrix-vector multiplications vs. timestep for various solvers for the fracture mechanics problem without preconditioning.*

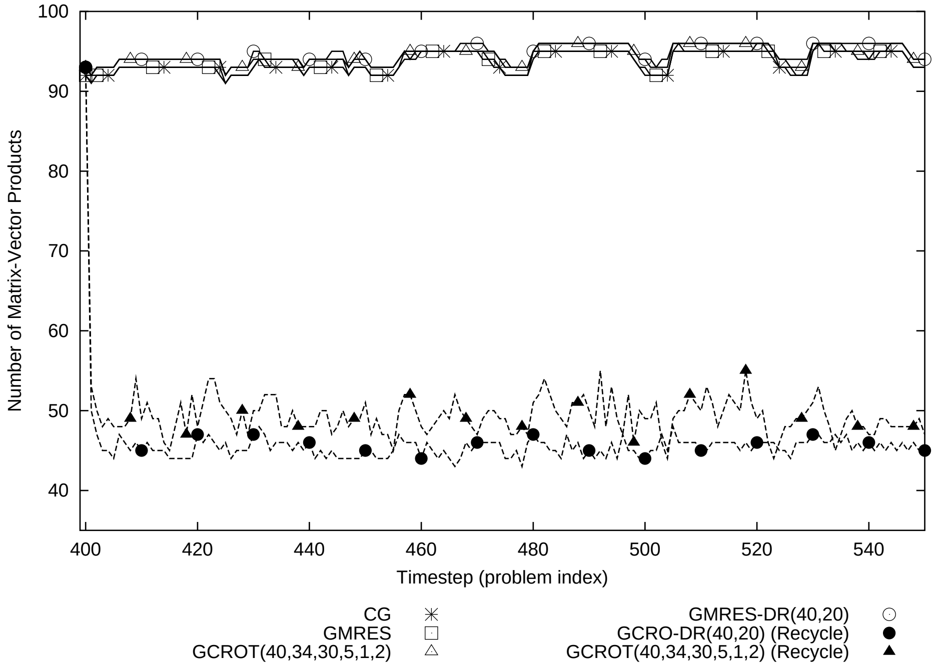
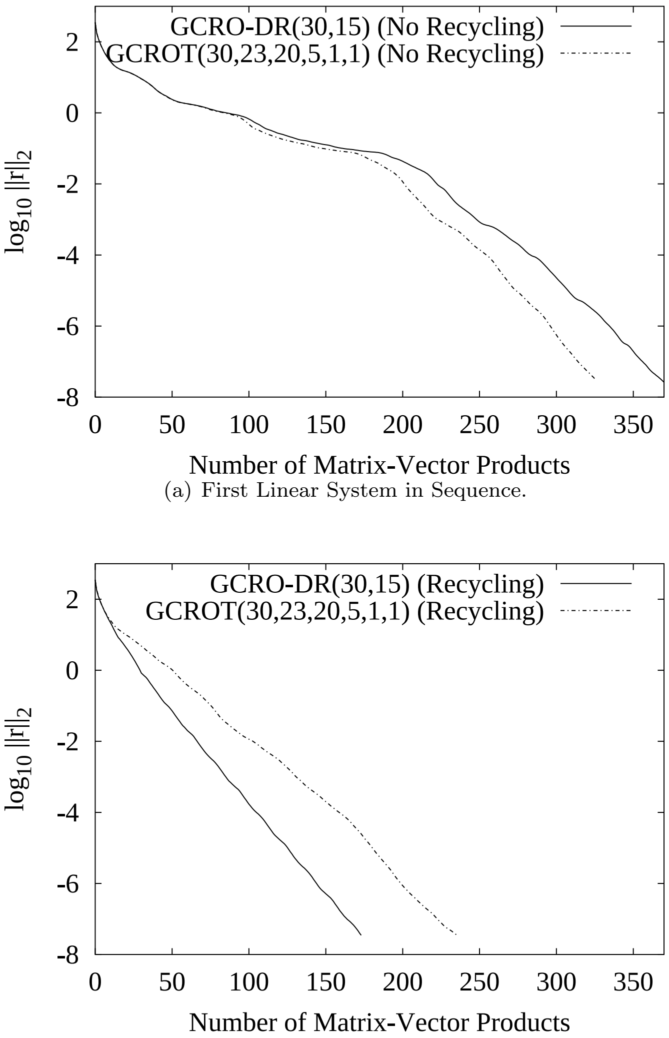


FIG. 4.2. *Number of matrix-vector multiplications vs. timestep for various solvers for the fracture mechanics problem with incomplete Cholesky preconditioning.*



(a) First Linear System in Sequence.

(b) Later Linear System in Sequence.

FIG. 4.3. *Typical convergence curves for GCROT and GMRES-DR applied to the fracture mechanics problem, with and without Krylov subspace recycling. The subspace recycled by GCRO-DR converges to an invariant subspace, whereas GCROT recycles the subspace selected in the last cycle of the previous linear system. This subspace may not be as important for the first cycle of the next system.*

which there is no recycled subspace available. For the first system, GCROT outperforms GCRO-DR. GCRO-DR and GCROT outperform the solvers without subspace recycling by a significant number of matrix-vector products. Overall, GCROT (without recycling) and CG show about the same convergence. Full GMRES outperforms CG, indicating that the convergence of CG is delayed due to effects of finite-precision arithmetic.

For the preconditioned case shown in Figure 4.2, GCRO-DR performs best, with GCROT with subspace recycling a close second. All the other solvers cluster near GMRES.

Comparing GMRES-DR and GCRO-DR, we see a significant difference in convergence, even though both methods focus on removing the same approximate eigenspace. The difference is due solely to subspace recycling. With no subspace to recycle, GCRO-DR is algebraically equivalent to GMRES-DR. The data suggests that the eigenspace associated with the interior eigenvalues is hard to estimate accurately, and GCRO-DR exhibits superior performance (except for the first system) because it does not have to recompute that space with each new linear system. Deflating the eigenspace associated with the 20 smallest eigenvalues is particularly well-suited to these problems because the matrices are SPD, and so the convergence is determined by the spectra. In Figure 4.3(a), we show typical convergence curves for GCRO-DR and GCROT without preconditioning for the first linear system in a sequence, when no subspace is available to recycle. At each cycle, GCROT continually updates the subspace it retains between cycles, whereas the subspace retained by GCRODR between cycles converges to an invariant subspace. Commonly, we have observed GCROT to outperform deflation-based solvers in the absence of Krylov subspace recycling. In Figure 4.3(b) we show typical convergence curves for GCRO-DR and GCROT for a later system in the sequence, when both methods use Krylov subspace recycling. The subspace recycled by GCRO-DR is nearly invariant, and GCRO-DR shows good convergence. The subspace retained by GCROT is the subspace that was selected in the last cycle of the previous linear system. This subspace may not be as important for the first cycle in the next linear system. This observation suggests that retaining the subspace determined through optimal truncation in the *first* cycle of the previous system may prove more beneficial than retaining the one determined in the last cycle of the previous system. This remains to be explored.

4.2. Electronic Structure. We consider a small model problem that arises in the KKR method [17, 16]. 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. We use 16 basis functions per atom, which leads to 864 unknowns. The matrix has about $300,000$ nonzeros. However, for increasingly larger systems the matrix 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. In particular, we solve for the first 32 unit Cartesian basis vectors corresponding to the 2×16 basis functions associated with the first two atoms. We give the convergence history for the first atom in Figure 4.4. 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. As a model problem we use the matrix `conf5.0_00l4x4.1000.mtx` downloaded from the Matrix-Market website at NIST [2]. The model problems were submitted by Björn Medeke (Dept. of Mathematics, University of Wuppertal) [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 4.5.

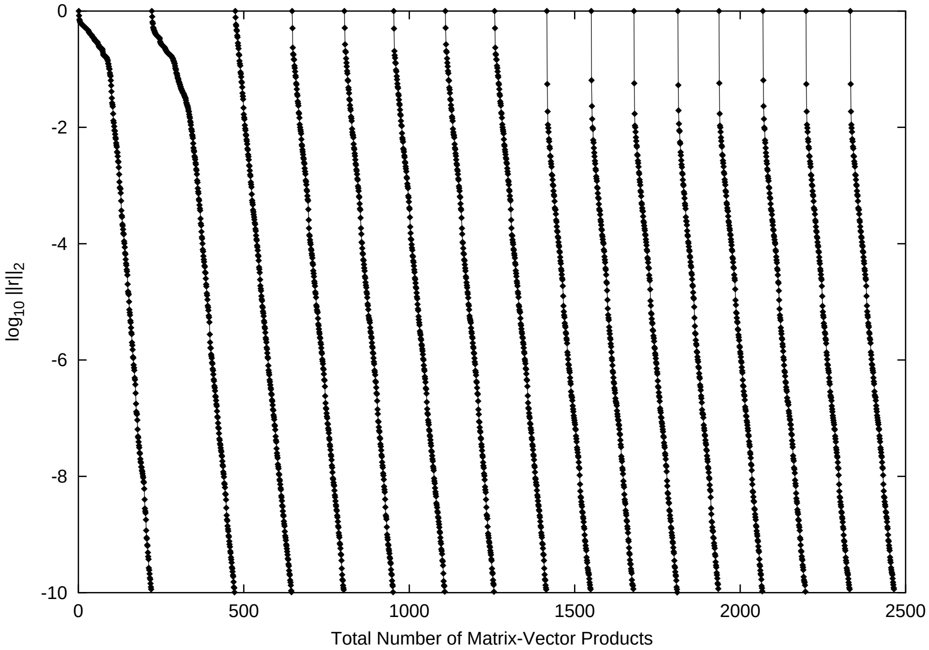


FIG. 4.4. *Convergence for 16 consecutive right hand sides for a small electronic structure problem. Each distinct curve gives the convergence for a subsequent right hand side, plotted against the total number of matrix-vector products. The first two right hand sides together take about 500 iterations, while the remaining right hand sides take about 140 iterations each, a reduction of almost 50%.*

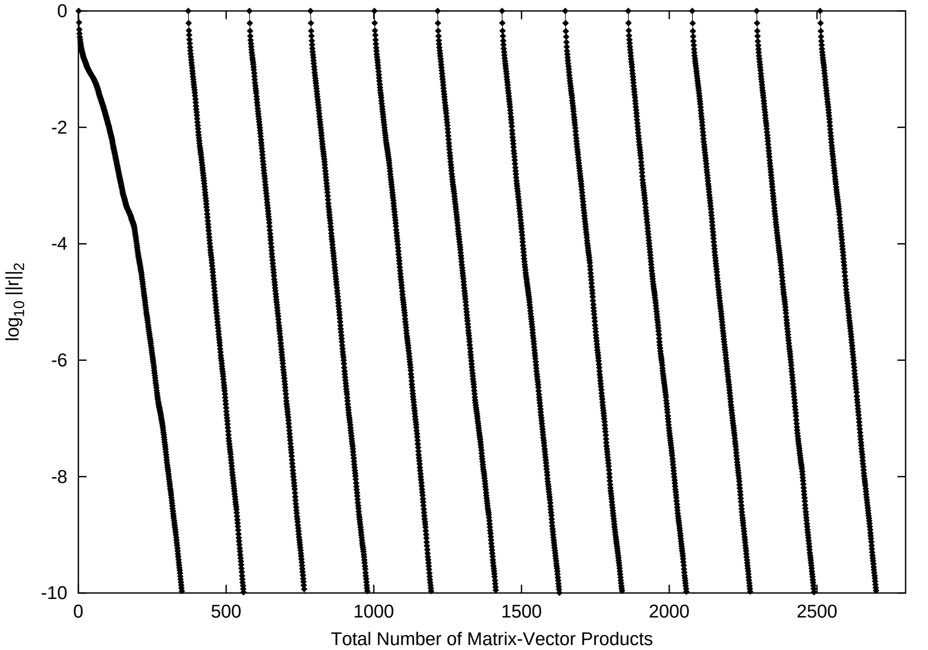


FIG. 4.5. *Convergence for 12 consecutive right hand sides for a model QCD problem from the NIST Matrix Market. Each distinct curve gives the convergence for a subsequent right hand side, plotted against the total number of matrix-vector products.*

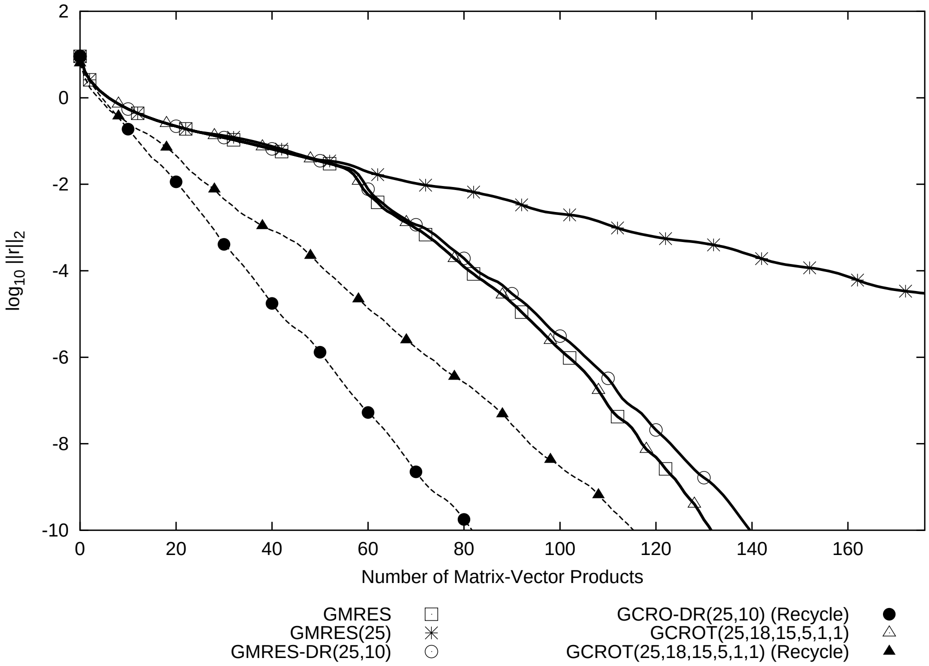


FIG. 4.6. *Number of matrix-vector products vs. timestep for various solvers for the convection-diffusion problem with* ***c=0****.*

4.4. Convection Diffusion. 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. When GCRO-DR keeps the same subspace between cycles as GMRES-DR, these methods are equivalent, so we do not plot the first run of GCRO-DR. The results for the **c=40** (nonsymmetric) case are quite interesting, and counterintuitive. The results are shown in Figure 4.6 for the **c=0** (symmetric) case and Figure 4.7 for the **c=40** (nonsymmetric) case. 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 residual to 1.0e−10.

For the **c=0** case, we see that the second runs of GCRO-DR and GCROT both converged faster than GMRES. All other solvers are, of course, slightly worse than GMRES, with GMRES(25) being far worse. 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 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 convergence was approximately the same for both runs. This means that the subspace

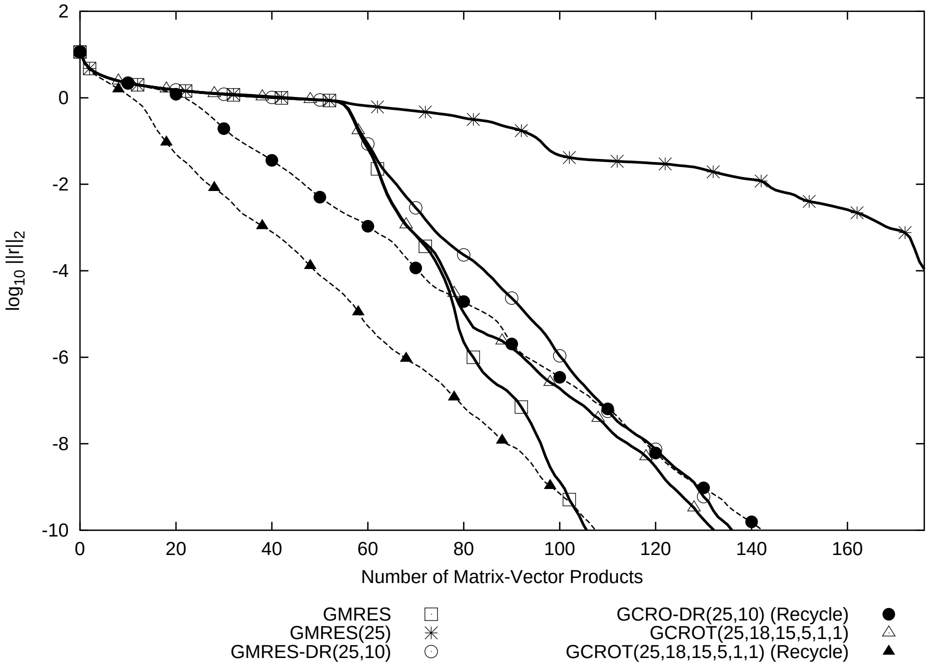


FIG. 4.7. *Number of matrix-vector products vs. timestep for various solvers for the convection-diffusion problem with* ***c=40****.*

Cosines of the principal angles between the recycled subspace and the subspace spanned by the 10 smallest eigenvectors		Cosines of the principal angles between the recycled subspace and the subspace spanned by the 21 smallest eigenvectors	
c=0	**c=40**	**c=0**	**c=40**
1.00000000000000	1.00000000000000	1.00000000000000	1.00000000000000
1.00000000000000	0.99999999999997	1.00000000000000	1.00000000000000
1.00000000000000	0.99999999839942	1.00000000000000	1.00000000000000
1.00000000000000	0.99999970490203	1.00000000000000	0.99999999999937
0.99999999999703	0.99990149788562	1.00000000000000	0.99999999545394
0.00000000593309	0.98844658524616	1.00000000000000	0.99999681064565
0.00000000003840	0.89957454665058	0.99999999999988	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

TABLE 4.2

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.*

it recycled failed to improve convergence.

Table 4.2 shows the cosines of the principal angles between the subspace recycled by GCRO-DR and the invariant subspace 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 recycled subspace for the **c=0** case only captures 5 eigenvectors. We choose to 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 solver GCRO-DR to implement Krylov subspace recycling of approximate invariant subspaces for Hermitian and non-Hermitian systems. When solving a sequence of linear systems, methods employing Krylov subspace recycling frequently outperformed GMRES while keeping only a small number of vectors. However, as the examples in section 4.4 show, this is not always the case. It is not yet well understood precisely how subspace selection affects convergence, so further theory is needed. Optimized minimum-residual methods for the Hermitian case are being developed.

Appendix. GCRO with Deflated Restarting (GCRO-DR).

```
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 $i = 1$.
if $\widetilde{Y}_k$ is defined (from solving a previous linear system) then
    Let $[Q, R]$ be the reduced QR-factorization of $A\widetilde{Y}_k$.
    $C_k = Q$
    $U_k = \widetilde{Y}_k R^{-1}$
    $x_1 = x_0 + U_k C_k^H r_0$
    $r_1 = r_0 - C_k C_k^H r_0$
else
    $v_1 = r_0/\|r_0\|_2$
    $c = \|r_0\|_2 e_1$
    Perform $m$ steps of GMRES, solving $min\|c - \overline{H}_m y\|_2$ for $y$ and generating $V_{m+1}$ and $\overline{H}_m$.
    $x_1 = x_0 + V_m y$
    $r_1 = V_{m+1}(c - \overline{H}_m y)$
    Compute the $k$ smallest eigenvectors $\widetilde{z}_j$ of $(H_m + h_{m+1,m}^2 H_m^{-H} e_m e_m^H)\widetilde{z}_j = \widetilde{\theta}_j \widetilde{z}_j$ and store in $P_k$.
    $\widetilde{Y}_k = V_m P_k$
    Let $[Q, R]$ be the reduced QR-factorization of $\overline{H}_m P_k$.
    $C_k = V_{m+1} Q$
    $U_k = \widetilde{Y}_k R^{-1}$
end if
while $\|r_i\|_2 > tol$ do
    $i = i + 1$
    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 $V_{m-k+1}$, $\overline{H}_{m-k}$, and $B_{m-k}$.
    Let $D_k$ be a diagonal scaling matrix such that $\widetilde{U}_k = U_k D_k$ where the columns of $\widetilde{U}_k$ have unit norm.
    $\widehat{V}_m = [\widetilde{U}_k \;\; V_{m-k}]$
    $\widehat{W}_{m+1} = [C_k \;\; V_{m-k+1}]$
    $\overline{G}_m = \begin{bmatrix} D_k & B_{m-k} \\ 0 & \overline{H}_{m-k} \end{bmatrix}$
    Solve $min\|\widehat{W}_{m+1}^H r_{i-1} - \overline{G}_m y\|_2$ for $y$.
    $x_i = x_{i-1} + \widehat{V}_m y$
    $r_i = r_{i-1} - \widehat{W}_{m+1} \overline{G}_m y$
    Compute the $k$ smallest eigenvectors $\widetilde{z}_j$ of $\overline{G}_m^H \overline{G}_m \widetilde{z}_i = \widetilde{\theta}_i \overline{G}_m^H \widehat{W}_{m+1}^H \widehat{V}_m \widetilde{z}_i$ and store in $P_k$.
    $\widetilde{Y}_k = \widehat{V}_m P_k$
    Let $[Q, R]$ be the reduced QR-factorization of $\overline{G}_m P_k$.
    $C_k = \widehat{W}_{m+1} Q$
    $U_k = \widetilde{Y}_k R^{-1}$
end while
Let $\widetilde{Y}_k = U_k$ (for the next system)
```

REFERENCES

[1] J. Baglama, D. Calvetti, G. H. Golub, and L. Reichel, *Adaptively preconditioned GMRES algorithms*, SIAM Journal on Scientific Computing, 20 (1999), pp. 243–269.

[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, R. 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 Journal on Scientific Computing, 21 (1999), pp. 836–850.

[4] M. Creutz, *Quarks, Gluons, and Lattices*, Cambridge University Press, 1986.

[5] E. de Sturler, *Nested Krylov methods based on GCR*, Journal of Computational and Applied Mathematics, 67 (1996), pp. 15–41.

[6] ———, *Truncation strategies for optimal Krylov subspace methods*, SIAM Journal on Numerical Analysis, 36 (1999), pp. 864–889.

[7] J. J. Dongarra, I. S. Duff, D. C. Sorensen, and H. A. van der Vorst, *Numerical linear algebra for high-performance computers*, Society for Industrial and Applied Mathematics, Philadelphia, PA, 1998.

[8] M. Eiermann, O. G. Ernst, and O. Schneider, *Analysis of acceleration strategies for restarted minimal residual methods*, Journal of Computational and Applied Mathematics, 123 (2000), pp. 261–292.

[9] S. C. Eisenstat, H. C. Elman, and M. H. Schultz, *Variational iterative methods for nonsymmetric systems of linear equations*, SIAM Journal on Numerical Analysis, 20 (1983), pp. 345–357.

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

[11] C. Farhat and F.-X. Roux, *Implicit parallel processing in structural mechanics*, in Computational Mechanics Advances, J. T. Oden, ed., vol. 2 (1), North-Holland, 1994, pp. 1–124.

[12] P. F. Fischer, *Projection techniques for iterative solution of $Ax = b$ with successive right-hand sides*, Comp. Meth. in Appl. Mech, 163 (1998), pp. 193–204.

[13] G. H. Golub and C. F. V. Loan, *Matrix Computations*, Johns Hopkins University Press, Baltimore, MD, USA, third ed., 1996.

[14] 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.*, Computers and Structures, 79 (2001), pp. 553–575.

[15] D. Johnson, D. Nicholson, F. Pinski, B. Gyorffy, and G. Stocks, *Energy and pressure calculations for random substitutional alloys*, Physical Review B, 41 (1990), pp. 9701–9716.

[16] W. Kohn and N. Rostoker, *Solution of the Schrödinger equation in periodic lattices with an application to metallic lithium*, Physical Review, 94 (1954), pp. 1111–1120.

[17] J. Korringa, *On the calculation of the energy of a Bloch wave in a metal*, Physica, XIII (1947), pp. 392–400.

[18] G. Mackey, *Reusing Krylov subspaces for sequences of linear systems*, master's thesis, 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, *A restarted GMRES method augmented with eigenvectors*, SIAM Journal on Matrix Analysis and Applications, 16 (1995), pp. 1154–1171.

[21] ———, *Implicitly restarted GMRES and Arnoldi methods for nonsymmetric systems of equations*, SIAM Journal on Matrix Analysis and Applications, 21 (2000), pp. 1112–1135.

[22] ———, *GMRES with deflated restarting*, SIAM Journal on Scientific Computing, 24 (2003), pp. 20–37.

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

[24] C. Rey and F. Risler, *A Rayleigh-Ritz preconditioner for the iterative solution to large scale nonlinear problems*, Numerical Algorithms, 17 (1998), pp. 279–311.

[25] F. Risler and C. Rey, *On the reuse of Ritz vectors for the solution to nonlinear elasticity problems by domain decomposition methods*, Contemporary Mathematics, 218 (1998), pp. 334–340.

[26] ———, *Iterative accelerating algorithms with Krylov subspaces for the solution to large-scale nonlinear problems*, Numerical Algorithms, 23 (2000), pp. 1–30.

[27] Y. Saad, *Analysis of augmented Krylov subspace methods*, SIAM Journal on Matrix Analysis and Applications, 18 (1997), pp. 435–449.

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

[29] Y. Saad, M. Yeung, J. Erhel, and F. Guyomarc'h, *A deflated version of the Conjugate Gradient algorithm*, SIAM Journal on Scientific Computing, 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. Szyld, *On the superlinear convergence of exact and inexact Krylov subspace methods*, Technical Report 03-3-13, Temple University, March 2003.

[32] A. Smirnov and D. Johnson, *Accuracy and limitations of localized Green's function methods for materials science applications*, Physical Review B, 64 (2001), pp. 235129–1 – 235129–9.

[33] J. van den Eshof, A. Frommer, T. Lippert, K. Schilling, and H. A. van der Vorst, *Numerical methods for the QCD overlap operator: I sign function and error bounds*, Comput. Phys. Commun., 146 (2002), pp. 203–224.

[34] B. Vital, *Etude de quelques méthodes de résolution de problèmes linéaires de grande taille sur multiprocessor*, PhD thesis, Université de Rennes I, Rennes, Nov 1990.

[35] U. M. Yang and K. A. Gallivan, *A new family of block methods*, Applied Numerical Mathematics: Transactions of IMACS, 30 (1999), pp. 155–173.

[36] R. Zeller, P. Dederichs, B. Ujfalussy, L. Szunyogh, and P. Weinberger, *Theory and convergence properties of the screened Korringa-Kohn-Rostoker method*, Physical Review B, 52 (1995), pp. 8807–8812.