THE ITERATIVE SOLUTION OF A SEQUENCE OF LINEAR SYSTEMS ARISING FROM NONLINEAR FINITE ELEMENT ANALYSIS

BY

MICHAEL LAWRENCE PARKS

B.S., Virginia Polytechnic Institute and State University, 1998
B.S., Virginia Polytechnic Institute and State University, 1998
M.S., Virginia Polytechnic Institute and State University, 2000

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Computer Science
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2005

Urbana, Illinois

To Nancy

Acknowledgments

I am indebted to my advisor, Eric de Sturler, for guiding both my research and my professional development. Throughout my time at Illinois, he provided encouragement, sound advice, good teaching, good company, and lots of good ideas.

I want to thank my committee members for their contributions to my graduate education. Additionally, I'm grateful to have been a participant in Mike Heath's 491 seminar, which covered many topics not taught in any ordinary class. I also thank Bob Dodds and Philippe Geubelle for providing the motivation for much of this thesis, and for supplying numerous "real-world" problems.

I benefitted greatly from interactions with my fellow graduate students, especially Chris Siefert, Ryan Szypowski, Bill Cochran, Greg Mackey, Rebecca Hartman-Baker, Hanna VanderZee, Vanessa Lopez, Naomi Caldwell, Zhen Cheng and Shun Wang. The ever-present comradery among graduate students in the numerical analysis group made Illinois a fun place to be and a difficult place to leave.

I'm grateful for financial support from Eric de Sturler, Bob Dodds, and the Computational Science and Engineering program, in the form of two Computational Science and Engineering fellowships. The CSE program also provided essential administrative support and computational facilities throughout my time at Illinois. This work was also supported in part by an NSF Combined Research and Curriculum Development Grant in Computational Materials Science and Nanoscale Science and Engineering, EEC-88101.

Finally, I thank my wife Nancy for her patience through four long Illinois winters.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

CG Conjugate Gradient Method.

FETI Finite Element Tearing and Interconnecting Method.

FGMRES Flexible GMRES.

GCR Generalized Conjugate Residual Method.

GCRO GCR with inner Orthogonalization.

GCRO-DR GCRO with Deflated Restarting.

GCROT GCRO with Optimal Truncation.

GMRES Generalized Minimum Residual Method.

GMRES-DR GMRES with Deflated Restarting.

GMRES-E GMRES with Eigenvectors.

GMRES-IR GMRES with Implicit Restarting.

GMRESR Recursive GMRES.

KKT Karush-Kuhn-Tucker.

PCG Preconditioned CG.

PCPG Preconditioned Conjugate Projected Gradient Method.

SPD Symmetric Positive Definite.

SPSD Symmetric Positive Semidefinite.

Chapter 1

Introduction and Motivation

Research on failure mechanisms (e.g. fatigue and fracture) of engineering components often focuses on modeling complex, nonlinear response. The analysis by finite element methods requires large-scale, very refined 3D solid models [19], which necessitate parallel computation. 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 large linear systems, thousands of times, to accomplish a realistic analysis. The equations generally remain symmetric positive definite but become very ill-conditioned due to localized damage (cracks) in the models. This leads to intolerably slow convergence of iterative methods. As a result, sparse direct solvers dominate commercial finite element software [16]. However, rapid advances in technology have caused a dramatic growth in the size of the linear systems to be solved. The fill generated by direct methods makes the storage requirements for such linear systems prohibitively expensive, and iterative solvers become the only viable option. While much work has been done on improving iterative solver and preconditioner technology for this class of problems, the primary focus has been on improving the convergence of individual linear systems. Improvements designed to accelerate the solution of a sequence of linear systems, taken as a whole, remain relatively unexplored. Herein, we develop enhancements for solvers and preconditioners that leverage work done solving previous linear

systems to improve the convergence of subsequent systems.

One avenue to improve the convergence of linear solvers considers more intelligent Krylov subspace methods. The standard, optimal iterative solver for symmetric positive definite (SPD) systems, the preconditioned conjugate gradient method (PCG), may in practice fail to converge for very ill-conditioned problems. The convergence is delayed or fails entirely due to a loss of orthogonality of the residual vectors in the PCG iteration and hence a rapid recurrence of certain eigenvectors [49] caused by finite precision computation. Optimal solvers for non-symmetric problems, such as the generalized minimum residual method (GMRES) [40], converge, but have much higher memory requirements, and may incur a cost comparable to that of a direct method [24]. We seek robust and efficient new Krylov methods capable of accelerating the convergence of a sequence of linear systems.

In Chapter 2 we investigate restarted Krylov subspace methods that retain a carefully chosen subspace between restarts in an effort to approximate the robustness of GMRES with the efficiency of PCG. These methods exploit the often relatively slow change in the linear systems from one timestep to the next by "recycling" Krylov subspace information from previous linear systems to accelerate convergence. Although motivated by problems in fracture mechanics, the solver techniques introduced here have proven effective for a wide variety of problems. In Chapter 3 we analyze the convergence of GCRO-DR, a solver introduced in Chapter 2 that recycles approximate invariant subspaces.

Improved preconditioners are also required to address the deficiencies of iterative solvers for this very important problem class. Domain decomposition methods based on substructuring have been applied successfully to many engineering problems. For the work of a domain decomposition method to scale (nearly) linearly with the number of subdomains, the method must employ some form of "coarse-space" preconditioner that employs ideas motivated by multigrid-type methods [47]. The finite element tearing and interconnecting

(FETI) [15, 16] method has the desirable property of showing scalability with respect to both the mesh and subdomain sizes. This comes at the cost of an expensive subproblem, as the inverse of a Schur complement matrix is required. We observe that the FETI method generates a Karush-Kuhn-Tucker (KKT) system. In other fields requiring the solution of KKT systems, it is more common to approximate inverses of Schur complements. In an effort to amortize the cost of this expensive subproblem, a factorized Schur complement matrix could be "recycled" for the next linear system, and used as an approximation to the true Schur complement matrix for that linear system. In Chapter 4 we show new connections between preconditioners and solvers for KKT systems and the original FETI method. These connections allow us to leverage existing work for KKT systems to create a framework for the analysis of improvements to the FETI method. We will provide eigenvalues bounds for FETI preconditioners, and develop a new FETI method that allows the use of an approximate Schur complement. Further, we generate eigenvalue bounds showing the potential impact on convergence caused by use of an approximate Schur complement.

Chapter 2

Recycling Krylov Subspaces for Sequences of Linear Systems

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 consider 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.

2.1 Introduction

We consider the solution of a sequence of general linear systems

$$A^{(i)}x^{(i)} = b^{(i)}, \qquad i = 1, 2, \ldots, \tag{2.1.1}$$

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 the systems 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 [19]. 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 [34, 35, 36]. 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 element tearing and interconnecting (FETI) method [16], 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 [5] 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 (2.1.1). When all right hand sides are available simultaneously, block methods such as block CG [32], block GMRES [52], and the family of block EN-like methods [53] are often suitable. However, block methods do not generalize to the case (2.1.1). If only one right hand side is available at a time, the method of Fischer [17], the deflated conjugate gradient method (deflated CG) [39], or the hybrid method of Simoncini and Gallopoulos [44] 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 (2.1.1), 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 [9]. We elaborate on the latter choice in section 2.2.3.

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 [41]. 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, 14]. When using exactly invariant subspaces, an augmentation approach is superior to a preconditioning approach [11]. Hence, we consider only the augmentation and orthogonalization approaches.

In section 2.2, 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 [31] and de Sturler's GCROT [9], 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 2.3, 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 2.4, we give the experimental results, which show that recycling can be very beneficial. Conclusions and future work are given in section 2.5.

2.2 Truncated and Augmented Krylov Methods

Restarting GMRES [40] 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 [11]. 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 [29] 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 [41]. 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 [29, 31]. 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 (2.1.1). 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 2.2.4. Because GMRES-DR cannot be used for Krylov subspace recycling, we combine ideas from GCRO [8] 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 2.2.5. In Chapter 3 we analyze the convergence of GCRO-DR.

Another strategy for subspace selection was proposed in [9] and was used for the GCROT method, an extension of GCRO. We discuss this approach, and its modification towards solving (2.1.1) in section 2.2.3.

We first review some definitions.

2.2.1 Definitions

In the following, we denote the range of a matrix $V_m \in \mathbb{C}^{n\times m}$ by $\mathcal{R}(V_m)$. The Arnoldi recurrence in GMRES leads to the following relation, which we denote as the Arnoldi relation.

$$AV_m = V_{m+1}\underline{H}_m, \tag{2.2.1}$$

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

$$Ay - \theta y \perp w, \qquad \forall w \in S. \tag{2.2.2}$$

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,

$$A^{-1}\widetilde{y} - \widetilde{\mu}\widetilde{y} \perp w \qquad \forall w \in AS, \tag{2.2.3}$$

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.

2.2.2 GMRES and GCR

We now review the linear solvers GMRES [40] and GCR [12], 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 (2.2.1) with $\mathcal{R}(V_m) = K^{(m)}(A,r_0)$. Then, we solve $\min \|c - \underline{H}_m d\|_2$ 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

$$\mathcal{R}(U_m) = K^{(m)}(A, r_0), \tag{2.2.4}$$

$$AU_m = C_m, \tag{2.2.5}$$

$$C_m^H C_m = I_m. \tag{2.2.6}$$

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 (2.2.5)-(2.2.6) still hold if $\mathcal{R}(U_m)$ is not a Krylov space, allowing us to find the minimum residual solution over any subspace $\mathcal{R}(U_m)$. In this case the method would not be called GCR, but the relations (2.2.5)-(2.2.6) are still valid.

2.2.3 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 [9, 25]. 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 (2.2.1).

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

$$AV_s = V_{s+1}\underline{H}_s. \tag{2.2.7}$$

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* [18, 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 (2.2.5)-(2.2.6). The minimum residual solution over $\mathcal{R}(U_k)$ 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 [8]. 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 (2.2.5)-(2.2.6) 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 [9].

GCROT can be modified to solve (2.1.1) 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 (2.2.5)-(2.2.6) 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 2.3.1. Moreover, even if we were to compute $A^{(i+1)}U_k^{old}$ directly, this can be faster than k separate matrix-vector multiplications [10]. 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.

2.2.4 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 the three linear solvers GMRES-E, GMRES-IR [30] 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

$$A[V_{m-k} \;\; \widetilde{Y}_k] = V_m \underline{H}_m, \tag{2.2.8}$$

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 (2.2.1). It can be shown that the augmented subspace

$$span\{r_0, Ar_0, A^2 r_0, \ldots, A^{m-k-1} r_0, \widetilde{y}_1, \widetilde{y}_2, \ldots, \widetilde{y}_k\} \tag{2.2.9}$$

is itself a Krylov subspace, but with another starting vector [30].

GMRES-DR is algebraically equivalent to GMRES-E at the end of each cycle if both select the same harmonic Ritz vectors. Because (2.2.9) 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 [30] that the subspace

$$span\{\widetilde{y}_1, \widetilde{y}_2, \ldots, \widetilde{y}_k, A\widetilde{y}_i, A^2\widetilde{y}_i, \ldots, A^{m-k}\widetilde{y}_i\} \tag{2.2.10}$$

is identical to subspace (2.2.9) 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

$$A[\widetilde{\Upsilon}_k \;\; V_{m-k}] = [\widetilde{\Upsilon}_k \;\; V_{m-k+1}] \underline{H}_m, \tag{2.2.11}$$

where $\underline{H}_m$ 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 (2.2.11), and finally restarts with

those vectors. Note that each column vector in V_{m-k} is orthogonal to $\mathcal{R}(\widetilde{Y}_k)$ in GMRES-DR, but this is not true in GMRES-E.

GMRES-DR cannot be directly used to solve (2.1.1), 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 [31] 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 (2.1.1). 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 (2.2.9) 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 2.3.1. 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 $\mathcal{R}(A^{(i+1)}\widetilde{Y}_k)$ and thus considers corrections in $\mathcal{R}(\widetilde{Y}_k)$ even though the residual is already orthogonal to $\mathcal{R}(A^{(i+1)}\widetilde{Y}_k)$. Although GMRES-E can be used when solving (2.1.1), because of these problems, we do not consider it further. Based on experiments, we believe

that it is preferable to preserve orthogonality to $\mathcal{R}(A^{(i+1)}\widetilde{Y}_k)$. The linear solver GCRO-DR, discussed next, accomplishes this.

2.2.5 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 [8]. The method is a generalization of GMRES-DR to solve (2.1.1). 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 (2.1.1) 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

$$A^{(i+1)}U_k = C_k, \tag{2.2.12}$$

$$C_k^H C_k = I_k, \tag{2.2.13}$$

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

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

2: $C_k = Q$

3: $U_k = \widetilde{Y}_k R^{-1}$

We find the optimal solution over the subspace $\mathcal{R}(U_k)$ 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

$$(I - C_k C_k^H)A^{(i+1)}V_{m-k} = V_{m-k+1}\underline{H}_{m-k}. \tag{2.2.14}$$

Each of the Arnoldi vectors $V_{m-k+1} = [v_1, v_2, \ldots, v_{m-k+1}]$ is orthogonal to $\mathcal{R}(C_k)$. We can rewrite (2.2.14) as

$$A[U_k \;\; V_{m-k}] = [C_k \;\; V_{m-k+1}] \begin{bmatrix} I_k & B_k \\ 0 & \underline{H}_{m-k} \end{bmatrix}, \tag{2.2.15}$$

where $B_k = C_k^H A V_{m-k}$. 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 (2.2.15) is not unnecessarily ill-conditioned. This improves the accuracy of the numerical solution.

We define

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

and write (2.2.15) more compactly, as

$$A\widehat{V}_m = \widehat{W}_{m+1}\underline{G}_m. \tag{2.2.16}$$

Note that $\underline{G}_m = \widehat{W}_{m+1}^H A \widehat{V}_m$ 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

$$t = \arg \min_{z \in \mathcal{R}(\widehat{V}_m)} \|r - Az\|_2, \tag{2.2.17}$$

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

$$\underline{G}_m y \cong \widehat{W}_{m+1}^H r = \|r\|_2 e_{k+1}, \tag{2.2.18}$$

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

$$r = r - A\widehat{V}_m y = r - \widehat{W}_{m+1}\underline{G}_m y, \tag{2.2.19}$$

$$x = x + \widehat{V}_m y. \tag{2.2.20}$$

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

$$\underline{G}_m^H \underline{G}_m \widetilde{z}_i = \widetilde{\theta}_i \underline{G}_m^H \widehat{W}_{m+1}^H \widehat{V}_m \widetilde{z}_i, \tag{2.2.21}$$

derived from (2.2.3), 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.

2.3 Test Problems

We discuss our main example in section 2.3.1, 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 non-symmetric 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 2.3.2, and a problem from lattice QCD in section 2.3.3. Finally, in section 2.3.4, 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 2.4.

2.3.1 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 [19].

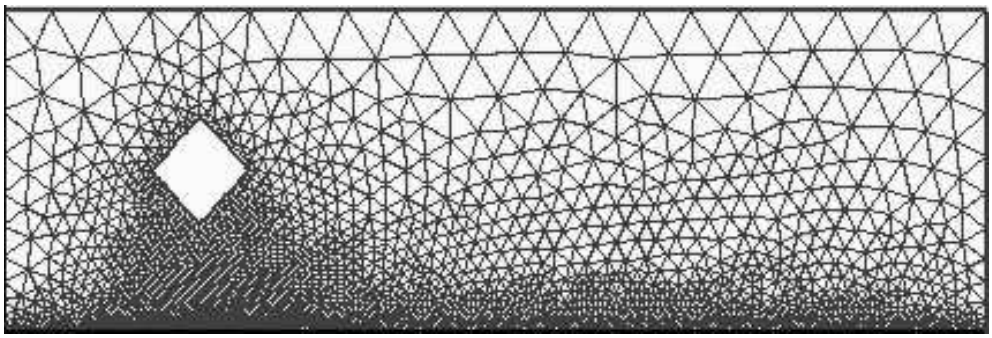


Figure 2.3.1: 2D plate mesh for crack propagation problem.

We study a sequence of linear systems taken from a finite element code developed by Philippe Geubelle and Spandan Maiti (both Aerospace Engineering, UIUC). The code simulates crack propagation in a metal plate using so-called "cohesive finite elements". The plate mesh is shown in Figure 2.3.1. The model is 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 2.4.1.

2.3.2 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 [23, 22].

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.

$$G^{ij} = t^i + t^i G_0^{ij} t^j + t^i G_0^{ik} t^k G_0^{kj} t^j + ..., \tag{2.3.1}$$

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,

$$\begin{aligned} G = t + tG_0(t + tG_0t + \ldots) &= t + tG_0G \Leftrightarrow \\ (I - tG_0)G &= t, \end{aligned} \tag{2.3.2}$$

where t is the block-diagonal matrix with blocks t^i. A localization strategy transforms (2.3.2) 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 [21, 54, 46]. We give results in Section 2.4.2, using a model problem provided by Duane Johnson (Materials Science and Engineering, 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.

2.3.3 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 [6, 51]. 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 [27]. 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 2.4.3.

2.3.4 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

$$u(x,0) = u(0,y) = 0,$$
$$u(x,1) = u(1,y) = 1.$$

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 2.4.4.

2.4 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 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 2.2.3) 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 [9, 25] 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.

2.4.1 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 2.4.1 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 2.4.1 minimize over a subspace of dimension 40. GMRES(40) is not shown in Figure 2.4.1 because it required an order of magnitude more matrix-vector multiplications than the other methods to converge. The results in Figure 2.4.2 are for the same sequence with

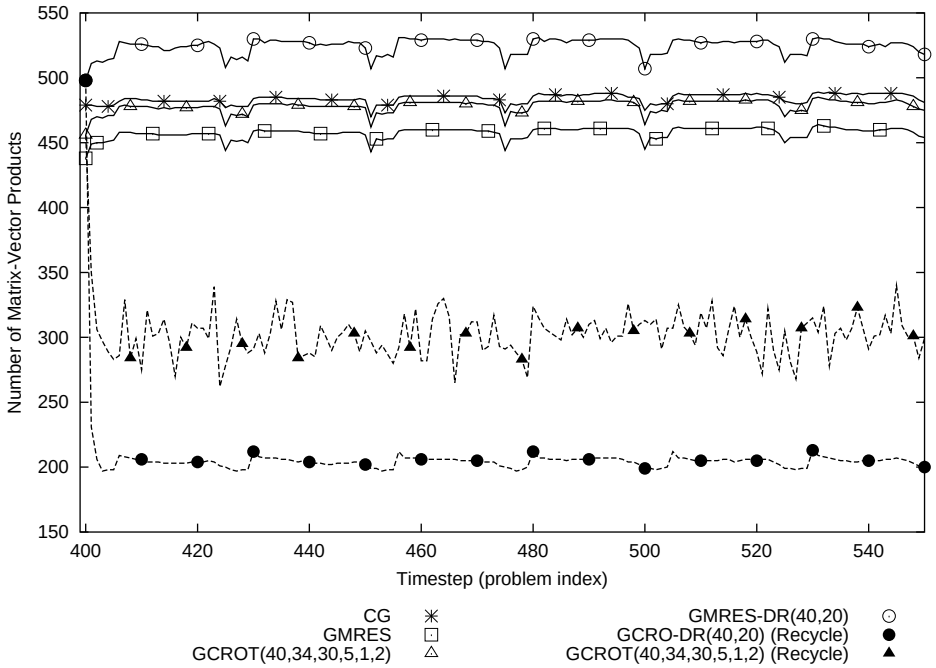


Figure 2.4.1: Number of matrix-vector multiplications vs. timestep for various solvers for the fracture mechanics problem without preconditioning.

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 2.4.1.

We see in Figure 2.4.1 that GCRO-DR, which employs subspace recycling, requires the fewest matrix-vector products, except for the first system in the sequence, for 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 2.4.2, GCRO-DR performs best, with GCROT with subspace recycling a close second. All the other solvers cluster near GMRES.

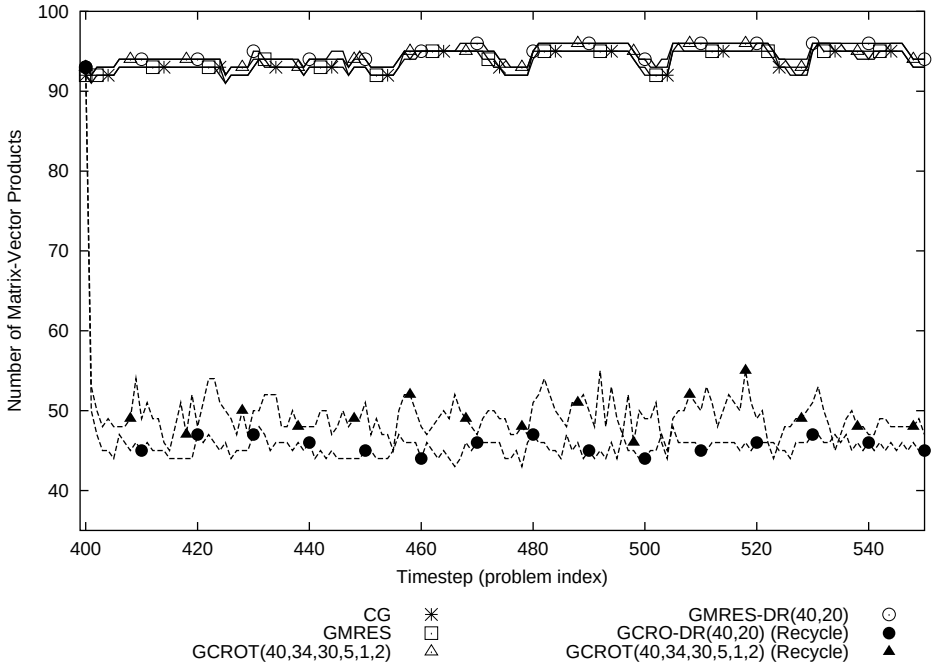
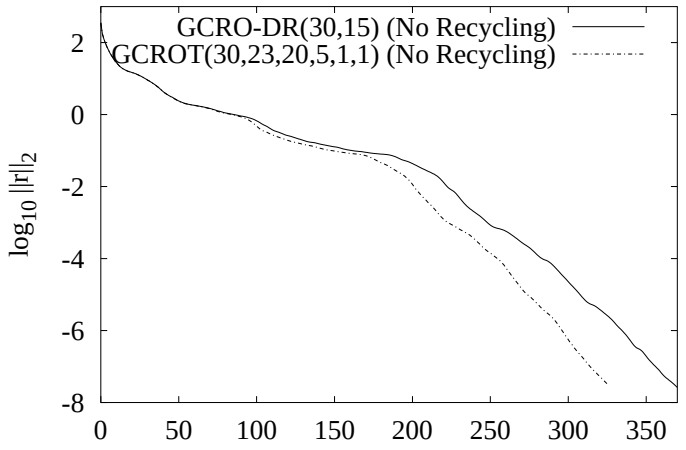


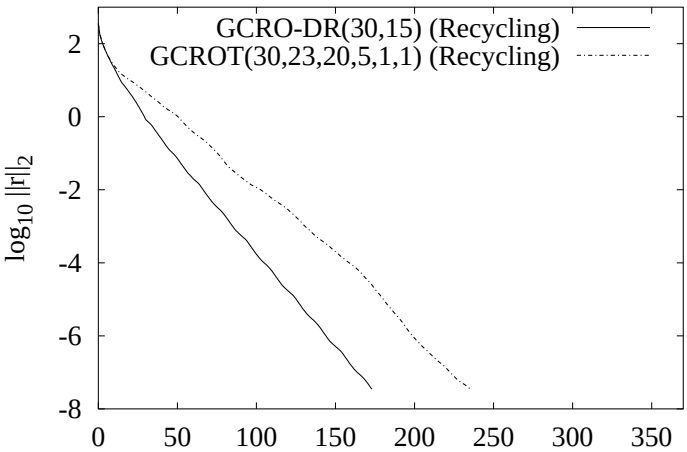
Figure 2.4.2: Number of matrix-vector multiplications vs. timestep for various solvers for the fracture mechanics problem with incomplete Cholesky preconditioning.

Table 2.4.1: 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**



(a) First Linear System in Sequence.



(b) Later Linear System in Sequence.

Figure 2.4.3: 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.

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 2.3(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 2.3(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.

2.4.2 Electronic Structure

We consider a small model problem that arises in the KKR method [23, 22]. 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 2.4.4. 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.

2.4.3 QCD

As a model problem we use the matrix "conf5.0_00l4x4.1000.mtx" downloaded from the Matrix-Market website at NIST [4]. The model problems were submitted by Björn Medeke (Dept. of Mathematics, University of Wuppertal) [27]. 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 2.4.5.

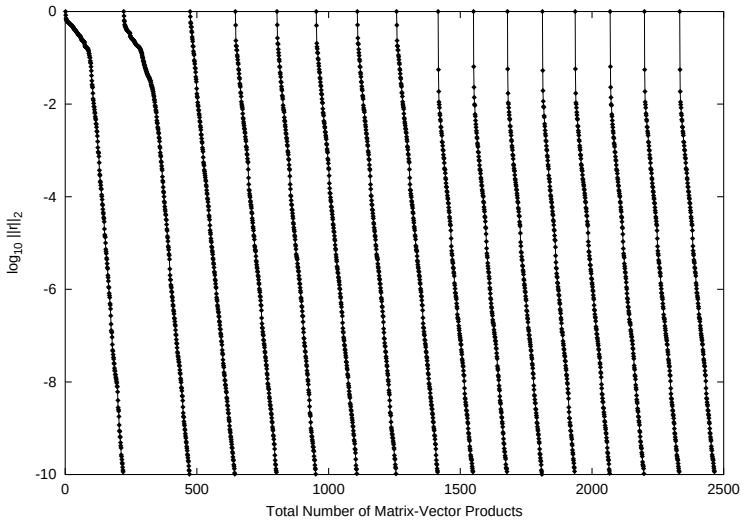


Figure 2.4.4: Convergence for 16 consecutive right hand sides for a small electronic structure problem. Each distinct curve gives the convergence for a subsequent right hand side, plotted against the total number of matrix-vector products. The first two right hand sides together take about 500 iterations, while the remaining right hand sides take about 140 iterations each, a reduction of almost 50%.

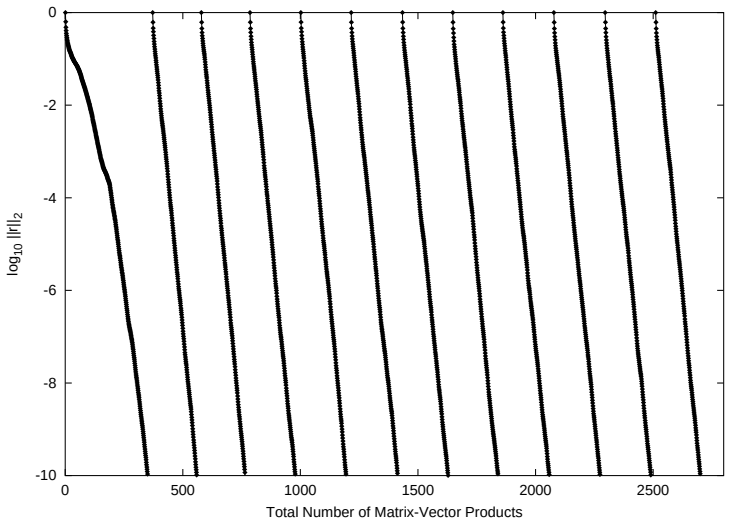


Figure 2.4.5: Convergence for 12 consecutive right hand sides for a model QCD problem from the NIST Matrix Market. Each distinct curve gives the convergence for a subsequent right hand side, plotted against the total number of matrix-vector products.

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 2.4.2: 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.

2.4.4 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 2.4.6 for the $c = 0$ (symmetric) case and Figure 2.4.7 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 con-

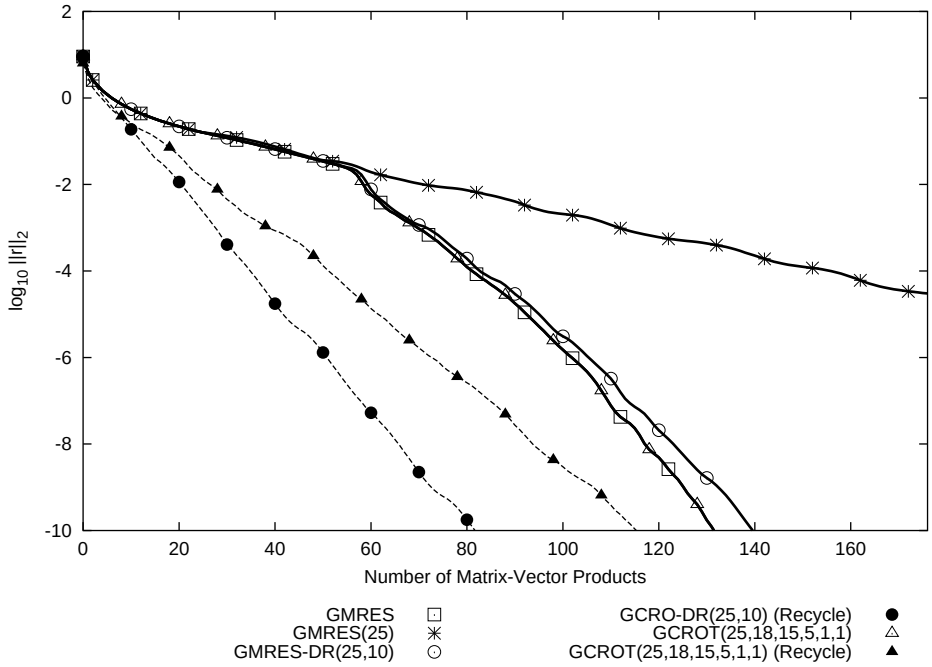


Figure 2.4.6: Number of matrix-vector products vs. timestep for various solvers for the convection-diffusion problem with $c = 0$.

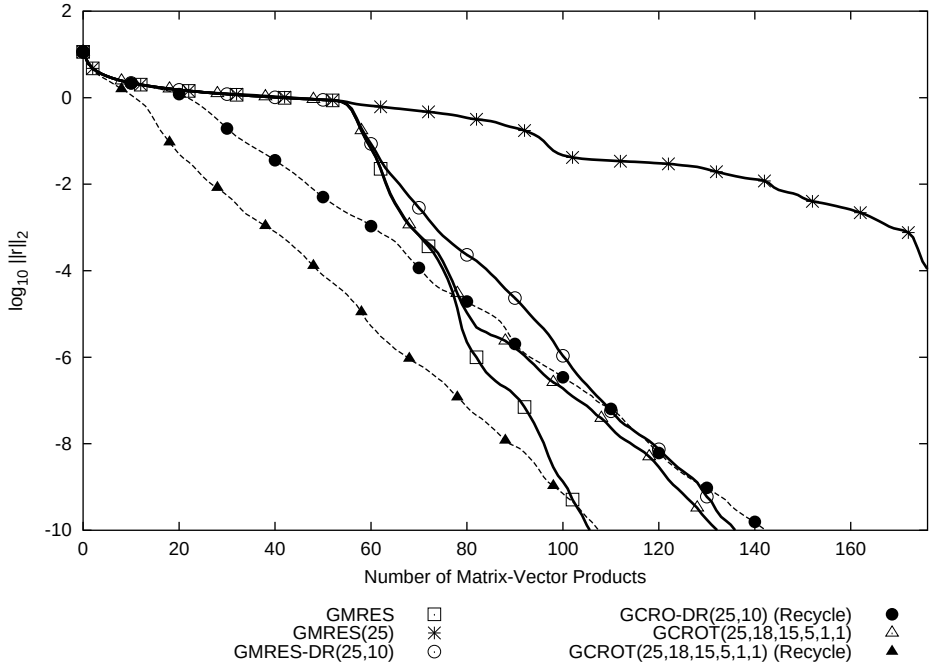


Figure 2.4.7: Number of matrix-vector products vs. timestep for various solvers for the convection-diffusion problem with $c = 40$.

verged 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 it recycled failed to improve convergence.

Table 2.4.2 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.

2.5 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 2.4.4 show, this is not always the case. It is not yet well understood precisely how subspace selection affects convergence. In Chapter 3 we examine this process further, and develop convergence bounds for GCRO-DR.

Chapter 3

Analysis of Krylov Subspace Recycling for Sequences of Linear Systems

In this chapter, we analyze the convergence of GCRO-DR, which recycles nearly invariant subspaces. We establish a bound on the residual norm produced by GCRO-DR in terms of GMRES with a deflated Krylov subspace and deflated initial residual vector. It is frequently suggested that deflating away eigenvalues closest to the origin is a desirable goal. Experimental and theoretical results show that while recycling invariant subspaces can be beneficial, better choices exist. In particular, we demonstrate that deflating away eigenvalues closest to the origin is not always best.

3.1 Introduction

In Chapter 2, the linear solver GCRO-DR was proposed for solving a sequence of general linear systems where only one linear system is available at a time, and both the matrix and right hand side change from one system to the next. Here, we take a theoretical look at its convergence properties. In section 3.2 we define some useful notation and relationships, and review Krylov subspace recycling. In section 3.3.1 we derive a convergence bound for GCRO-DR, and in section 3.3.2 we show several experiments illustrating the conditions under which recycling nearly invariant subspaces is beneficial. We offer conclusions in

section 3.4.

3.2 Some Notation and Useful Relationships

Here, we introduce notation that will be valuable for our later analysis, catalog useful relationships between various projectors, and review the concept of the "one-sided distance" between subspaces. We assume the notation and discussion on Krylov subspace recycling from Chapter 2.

We denote the range of a matrix $C_k \in \mathbb{C}^{n\times k}$ by $\mathcal{R}(C_k)$. $C_k^\dagger$ denotes the Moore-Penrose pseudoinverse of C_k, and Π_C denotes the orthogonal projector onto $\mathcal{R}(C_k)$. Let $\mathcal{R}(Q_\ell)$ denote an ℓ-dimensional simple invariant subspace of the matrix $A \in \mathbb{C}^{n\times n}$. We use P_Q to represent the spectral projector [48, §V.1.3] onto $\mathcal{R}(Q_\ell)$. Since a projector acts as the identity over its range, we see that

$$\Pi_Q P_Q = P_Q. \tag{3.2.1}$$

We define the *one-sided distance* from the subspace $\mathcal{R}(Q_\ell)$ to the subspace $\mathcal{R}(C_k)$ $(k \geq \ell)$, as

$$\delta(Q_\ell, C_k) \equiv \| (I - \Pi_C)\Pi_Q \|_2. \tag{3.2.2}$$

$\delta(Q_\ell, C_k)$ is equal to the sine of the largest principal angle between $\mathcal{R}(Q_\ell)$ and $\mathcal{R}(C_k)$ [2]. This means that any unit vector in $\mathcal{R}(Q_\ell)$ has a component of at most length δ orthogonal to $\mathcal{R}(C_k)$. To the extent that these two subspaces coincide, $\delta(Q_\ell, C_k)$ approaches zero, and $\delta(Q_\ell, C_k) = 0$ if and only if $\mathcal{R}(Q_\ell) \subseteq \mathcal{R}(C_k)$. In particular, we will be considering the case where a subspace $\mathcal{R}(C_k)$ contains (or nearly contains) an invariant subspace $\mathcal{R}(Q_\ell)$.

3.3 Analysis of Deflation-Based Krylov Subspace Recycling

In section 3.3.1 we present a bound on the residual norm produced by GCRO-DR in terms of a GMRES process with a deflated Krylov space. We operate under the assumption that the recycled subspace $\mathcal{R}(C_k)$ contains or nearly contains an invariant subspace $\mathcal{R}(Q_\ell)$, where $k \geq \ell$, for some ℓ-dimensional invariant subspace $\mathcal{R}(Q_\ell)$ of A. All theoretical results require only that $\delta(Q_\ell, C_k) < 1$, but are more useful when $\delta(Q_\ell, C_k)$ is small. In section 3.3.2 we present some numerical experiments, and analyze them by applying the newly developed bounds. In particular, we show that recycling invariant subspaces can be quite effective on certain problems. However, we also demonstrate that deflating away eigenvalues closest to the origin is not always best, and when selecting a subspace to recycle, there exist better choices than invariant subspaces.

3.3.1 Recycling Invariant Subspaces: Theory

A deflated GMRES process is one in which all components from a particular invariant subspace have been removed from the individual residual, and the search subspace does not contain any components in this invariant subspace. The deflated problem can be expressed as

$$\min_{d \in AK^{(j)}\left(A,(I-P_Q)r_0\right)} \| (I-P_Q)\, r_0 - d \|_2, \tag{3.3.1}$$

where $r_0 = b - Ax_0$ is the residual for the initial guess x_0, and P_Q is the spectral projector onto an invariant subspace $\mathcal{R}(Q_\ell)$. In the eigenvector decomposition of A, the vector $(I-P_Q)\, r_0$ has no components in the subspace $\mathcal{R}(Q_\ell)$, and the same holds for all vectors in the j-dimensional Krylov subspace $K^{(j)}(A, (I-P_Q)\, r_0)$. In GCRO-DR, we do not

have available the invariant subspace $\mathcal{R}(Q_\ell)$ or the associated spectral projector P_Q, so instead we use the subspace $\mathcal{R}(C_k)$ and the associated orthogonal projector Π_C. Although $(I-P_Q)r_0$ has no components in $\mathcal{R}(Q_\ell)$, this is not necessarily true for $(I-\Pi_C)r_0$. As we cannot generate the Krylov subspace $AK^{(j)}(A,(I-P_Q)r_0)$, we instead consider the Krylov subspace $\mathcal{R}(V_j)$, where $V_j \in \mathbb{C}^{n\times j}$ is an orthonormal basis for

$$(I-\Pi_C)AK_j((I-\Pi_C)A,(I-\Pi_C)r_0). \tag{3.3.2}$$

We note that $(I-\Pi_C)V_j = V_j$. First, GCRO-DR finds the minimum residual solution over $\mathcal{R}(C_k)$, then updates the residual as $r_1 = (I-\Pi_C)r_0$, and finally computes the minimum residual solution over the subspace $\mathcal{R}(V_j)$. As such, we consider the related problem

$$\min_{d\in\mathcal{R}\left[(I-P_Q)V_j\right]} \|(I-P_Q)r_1 - d\|_2 \tag{3.3.3}$$

as an approximation to (3.3.1). In the eigenvector decomposition of A, the vector $(I-P_Q)r_1$ has no components in $\mathcal{R}(Q_\ell)$, and neither do the column vectors of $(I-P_Q)V_j$. Even though we recycle the subspace $\mathcal{R}(C_k)$, which contains an approximate invariant subspace, and use an orthogonal projector Π_C rather than a spectral projector, we show below that we can bound the convergence of GCRO-DR using the deflated problem (3.3.3), so long as $\delta(Q_\ell,C_k) < 1$. We will see below that if $\|P_Q\|_2$ is large, the bound may be loose.

Some of the following discussion was inspired by [45], which was in turn influenced by [41].

Theorem 3.3.1. *Let $\mathcal{R}(Q_\ell)$ be an ℓ-dimensional invariant subspace of $A \in \mathbb{C}^{n\times n}$. Let $\mathcal{R}(C_k)$ be a k-dimensional subspace $(k \geq \ell)$ such that $\delta(Q_\ell,C_k) < 1$. Let $\mathcal{R}(V_j)$ be a j-dimensional Krylov subspace, as defined in* (3.3.2)*. Let $r_0 \in \mathbb{C}^n$, and $r_1 = (I-\Pi_C)r_0$.*

Then,

$$\min_{d_1 \in \mathcal{R}\left([V_j, C_k]\right)} \|r_0 - d_1\|_2 \leq \min_{d_2 \in \mathcal{R}\left(V_j\right)} \{\|(I - P_Q)(r_1 - d_2)\|_2 + \gamma \|r_1 - d_2\|_2\}, \qquad (3.3.4)$$

where $\gamma = \|(I - \Pi_C)P_Q\|_2$.

Proof.

$$\begin{aligned}
\min_{d \in \mathcal{R}\left([V_j, C_k]\right)} \|r_0 - d\|_2 &= \min_{d, \widetilde{d} \in \mathcal{R}\left([V_j, C_k]\right)} \|r_1 - d - \widetilde{d}\|_2 \\
&= \min_{d, \widetilde{d} \in \mathcal{R}\left([V_j, C_k]\right)} \{\| (I - P_Q)(r_1 - d) + P_Q(r_1 - d) - \widetilde{d}\|_2\} \\
&\leq \min_{d \in \mathcal{R}\left(V_j\right)} \{\| (I - P_Q)(r_1 - d) + P_Q(r_1 - d) \\
&\quad -\Pi_C[(I - P_Q)(r_1 - d) + P_Q(r_1 - d)]\|_2\} \ {}^{1} \\
&= \min_{d \in \mathcal{R}\left(V_j\right)} \{\| (I - \Pi_C)(I - P_Q)(r_1 - d) + (I - \Pi_C)P_Q(r_1 - d)\|_2\} \\
&\leq \min_{d \in \mathcal{R}\left(V_j\right)} \{\| (I - \Pi_C)\|_2 \| (I - P_Q)(r_1 - d)\|_2 \\
&\quad + \| (I - \Pi_C)P_Q(r_1 - d)\|_2\} \\
&\leq \min_{d \in \mathcal{R}\left(V_j\right)} \{\| (I - P_Q)(r_1 - d)\|_2 + \gamma \|r_1 - d\|_2\}.
\end{aligned}$$

□

When considering Theorem 3.3.1, it is useful to think of a GCRO-DR process that has solved at least one linear system in the sequence, and is just beginning the first cycle on the next linear system in the sequence.

The term γ in Theorem 3.3.1 depends on the one-sided distance (3.2.2) between the invariant subspace $\mathcal{R}(Q_\ell)$ and the subspace $\mathcal{R}(C_k)$. We observe the following relationship.

[1] We have set $\widetilde{d} = \Pi_C[(I - P_Q)(r_1 - d) + P_Q(r_1 - d)]$.

Proposition 3.3.2. *Assume the notation from Theorem 3.3.1. Then,*

$$\begin{aligned}\gamma = \|(I-\Pi_C)P_Q\|_2 &= \|(I-\Pi_C)\Pi_Q P_Q\|_2 \\ &\leq \|(I-\Pi_C)\Pi_Q\|_2 \|P_Q\|_2 \\ &= \delta(Q_\ell, C_k)\,\|P_Q\|_2.\end{aligned}$$

Proof. The first inequality follows from (3.2.1). □

Again, we observe that any unit vector in $\mathcal{R}(Q_\ell)$ has at most a component of length δ orthogonal to $\mathcal{R}(C_k)$, and if $\mathcal{R}(Q_\ell) \subseteq \mathcal{R}(C_k)$, then $\delta = \gamma = 0$. If $\|P_Q\|_2$ is large, we must have δ small if γ is to be small.

Continuing with Theorem 3.3.1, let $d = V_j y$ for some $y \in \mathbb{R}^j$, and rewrite the bound in (3.3.4) as

$$\min_{y\in\mathbb{R}^j}\{\|(I-P_Q)r_1 - (I-P_Q)V_j y\|_2 + \gamma\|r_1 - V_j y\|_2\}. \tag{3.3.5}$$

We will use (3.3.5) to bound the convergence of GCRO-DR. Note that the first term in (3.3.5) is just the deflated problem (3.3.3), and the second term in (3.3.5) goes to zero as γ goes to zero. In this case, it is reasonable to think of the right term in (3.3.5) as a perturbation of the left term.

Proposition 3.3.3. *Assume the notation from Theorem 3.3.1. Then,*

$$\begin{aligned}&\min_{y\in\mathbb{R}^j}\{\|(I-P_Q)\left(r_1 - V_j y\right)\|_2 + \gamma\|r_1 - V_j y\|_2\} \leq \\ &\min_{y\in\mathbb{R}^j}\|(I-P_Q)r_1 - (I-P_Q)V_j y\|_2 + \gamma\|\left[(I-P_Q)V_j\right]^\dagger\|_2 \cdot \|P_Q\|_2 \cdot \|(I-\Pi_V)r_1\|_2.\end{aligned}$$

Proof. Let $y_b = \left[(I-P_Q)V_j\right]^\dagger (I-P_Q)\,r_1$ be the minimizing argument in (3.3.3). Clearly,

$$\begin{aligned} & \min_{y\in\mathbb{R}^j}\{\|\,(I-P_Q)\,r_1-(I-P_Q)\,V_j y\|_2+\gamma\|r_1-V_j y\|_2\} \\ \leq\ & \|\,(I-P_Q)\,r_1-(I-P_Q)\,V_j y_b\|_2+\gamma\|r_1-V_j y_b\|_2. \end{aligned}$$

Writing out the explicit representation for y_b gives

$$\begin{aligned} \|r_1-V_j y_b\|_2 &= \|\left(I-V_j\left[(I-P_Q)V_j\right]^\dagger (I-P_Q)\right)r_1\|_2 \\ &= \|\left(I-V_j\left[(I-P_Q)V_j\right]^\dagger (I-P_Q)\right)(I-\Pi_V)\,r_1\|_2 \qquad (3.3.6) \\ &\leq \|I-V_j\left[(I-P_Q)V_j\right]^\dagger (I-P_Q)\,\|_2\cdot\|\,(I-\Pi_V)\,r_1\|_2. \end{aligned}$$

Equation (3.3.6) follows from the observation that $V_j\left[(I-P_Q)V_j\right]^\dagger (I-P_Q)$ is an oblique projector onto $\mathcal{R}\left(V_j\right)$. It follows that

$$\begin{aligned} \|I-V_j\left[(I-P_Q)V_j\right]^\dagger (I-P_Q)\,\|_2 &= \|V_j\left[(I-P_Q)V_j\right]^\dagger (I-P_Q)\,\|_2 \\ &\leq \|V_j\|_2\cdot\|\left[(I-P_Q)V_j\right]^\dagger\|_2\cdot\|I-P_Q\|_2 \\ &= \|\left[(I-P_Q)V_j\right]^\dagger\|_2\cdot\|P_Q\|_2, \end{aligned}$$

where we have used the assumption that V_j has orthonormal columns. □

Finally, we bound $\|\left[(I-P_Q)V_j\right]^\dagger\|_2$.

Proposition 3.3.4. *Assume the notation from Theorem 3.3.1. For each $\mathcal{R}(Q_\ell)$ such that $\delta(Q_\ell,C_k)<1$,*

$$\|\left[(I-P_Q)V_j\right]^\dagger\|_2 \leq \frac{1}{1-\delta}.$$

Proof. We observe that

$$\| \left[(I - P_Q) V_j \right]^\dagger \|_2 \quad = \quad [\sigma_{min} \left(V_j - P_Q V_j \right)]^{-1},$$

where σ_{min} denotes the smallest singular value of a matrix. We now proceed to find a lower bound on $\|V_j z - P_Q V_j z\|_2$ over all $z \in \mathbb{R}^j$, $\|z\|_2 = 1$. We start by considering

$$P_Q V_j z = C_k \xi_1 + C_\perp \xi_2,$$

where $\xi_1 \in \mathbb{R}^k$, $\xi_2 \in \mathbb{R}^{n-k}$, C_k is an orthonormal basis for $\mathcal{R}(C_k)$, and $[C_k \, C_\perp]$ is unitary. We have expressed the vector $P_Q V_j z$ as the sum of its components in $\mathcal{R}(C_k)$ and $\mathcal{R}(C_\perp)$, for any unit vector z. It follows from the definition of $\delta(Q_\ell, C_k)$ that

$$\|C_\perp \xi_2\|_2 \leq \delta \|P_Q V_j z\|_2. \tag{3.3.7}$$

Inequality (3.3.7) implies that

$$\|C \xi_1\|_2 \geq \sqrt{1 - \delta^2} \, \|P_Q V_j z\|_2.$$

We have that

$$V_j z - P_Q V_j z = -C_k \xi_1 + V_j z - C_\perp \xi_2,$$

where

$$V_j z - C_\perp \xi_2 \in \mathcal{R}(C_\perp),$$

by construction. Thus,

$$\begin{aligned} \|V_j z - P_Q V_j z\|_2^2 &= \|C_k \xi_1\|_2^2 + \|V_j z - C_\perp \xi_2\|_2^2 \\ &\geq \left(1-\delta^2\right) \|P_Q V_j z\|_2^2 + \|V_j z - C_\perp \xi_2\|_2^2. \end{aligned}$$

We show that either one of these terms may be arbitrarily close to zero, but that the sum of the two terms together can always be bounded away from zero. We consider two cases, based on the size of $\alpha \equiv \|P_Q V_j z\|_2$.

Case I. $\alpha < 1$. In this case, $\left(1-\delta^2\right)\alpha^2$ may be small. However,

$$\begin{aligned} \|V_j z - P_Q V_j z\|_2^2 &\geq \|V_j z - C_\perp \xi_2\|_2^2 \\ &\geq (1-\delta\alpha)^2 \\ &\geq (1-\delta)^2, \end{aligned}$$

since $0 \leq \delta < 1$ and $\alpha < 1$.

Case II. $\alpha \geq 1$. In this case, $\|V_j z - C_\perp \xi_2\|_2$ may be zero. However,

$$\begin{aligned} \|V_j z - P_Q V_j z\|_2^2 &\geq \left(1-\delta^2\right)\alpha^2 \\ &\geq 1-\delta^2 \\ &\geq (1-\delta)^2, \end{aligned}$$

since $0 \leq \delta < 1$ and $\alpha \geq 1$. □

Remark 3.3.5. Proposition 3.3.4 shows that as soon as $\delta < 1$, we have an upper bound on $\| \left[(I - P_Q) V_j\right]^\dagger \|_2$. In all experiments conducted, we observed that $\|P_Q V_j\|_2 \ll 1$, and thus case I of Proposition 3.3.4 applies. In this situation, $\delta \approx 0$ and $\| \left[(I - P_Q) V_j\right]^\dagger \|_2 \approx 1$.

Corollary 3.3.6. *Assume the notation from Theorem 3.3.1. For each $\mathcal{R}(Q_\ell)$ such that*

$\delta(Q_\ell, C_k) < 1$,

$$\begin{aligned} \min_{d_1 \in \mathcal{R}[V_j, C_k]} \|r_0 - d_1\|_2 &\leq \min_{d_2 \in \mathcal{R}[(I-P_Q)V_j]} \|(I-P_Q)r_1 - d_2\|_2 \\ &+ \left(\frac{\gamma}{1-\delta}\right) \|P_Q\|_2 \cdot \|(I-\Pi_V)r_1\|_2. \end{aligned} \tag{3.3.8}$$

So, we see that the norm of the residual produced by GCRO-DR can be bounded above by (3.3.3), plus a term that approaches zero as $\mathcal{R}(Q_\ell)$ is increasingly contained in $\mathcal{R}(C_k)$ and as $\|(I-\Pi_V)r_1\|_2$ becomes small.

Note that (3.3.8) is true for any invariant subspace $\mathcal{R}(Q_\ell)$, but if this choice makes $\frac{\gamma}{1-\delta}\|P_Q\|_2$ very large, the bound may be very loose. It is therefore desirable to select the best $\mathcal{R}(Q_\ell)$ over all possible invariant subspaces $\mathcal{R}(Q_\ell)$ such that $\delta(Q_\ell, C_k) < 1$, where $\ell < k$, in general. When computing the bound (3.3.8) in section 3.3.2 for given values of ℓ and k, we will select the invariant subspace $\mathcal{R}(Q_\ell)$ that minimizes $\frac{\gamma}{1-\delta}\|P_Q\|_2$.

Corollary 3.3.7. *Assume the notation from Theorem 3.3.1, and that the matrix A is Hermitian. For each $\mathcal{R}(Q_\ell)$ such that $\delta(Q_\ell, C_k) < 1$,*

$$\begin{aligned} \min_{d_1 \in \mathcal{R}[V_j, C_k]} \|r_0 - d_1\|_2 &\leq \min_{d_2 \in \mathcal{R}[(I-\Pi_Q)V_j]} \|(I-\Pi_Q)r_1 - d_2\|_2 \\ &+ \frac{\delta}{1-\delta}\|(I-\Pi_V)r_1\|_2. \end{aligned}$$

Corollary 3.3.7 shows that the bound is tighter in the Hermitian case, which suggests that recycling invariant subspaces should be particularly effective for sequences of Hermitian systems.

3.3.2 Recycling Invariant Subspaces: Numerical Experiments

We show three example problems in this section. In the first example we examine a class of matrices with a parameter that controls the deviation from normality. This allows us to examine the influence of normality on the recycling process when invariant subspaces are used. The second example is a simple convection-diffusion problem. The final example concerns a matrix with a random eigenbasis but only ten distinct eigenvalues, and shows that a poorly chosen recycled subspace can severely harm convergence.

We will only consider the bound (3.3.8) over the first cycle, because we are primarily interested in how the recycling process impacts the initial convergence.

EXAMPLE 3.3.1. Here, we consider Example 4.4 from [45]. We use a set of 100×100 matrices $A^{(i)} = S^{(i)}\Lambda(S^{(i)})^{-1}$, $(i = 1,2,3)$ with $\kappa_1(S^{(1)}) = 1$, $\kappa_2(S^{(2)}) = 10^3$, $\kappa_3(S^{(3)}) = 10^6$. For each κ_i, the matrix $S^{(i)}$ is defined as $D^{(i)}U^*$, where $D^{(1)} = I$, and $D^{(j)} = diag(1 : \kappa_j/100 : \kappa_j)$ for $j = 2,3$. The matrix U is the orthogonal matrix in the QR factorization of the lower triangular part of

$$\begin{bmatrix} 1 & n+1 & 2n+1 & \cdots & \vdots \\ 2 & n+2 & 2n+2 & \cdots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ n & 2n & 3n & \cdots & n^2 \end{bmatrix}.$$

By construction, $\kappa(S^{(i)}) = \kappa_i$, $i = 1,2,3$. The matrix $\Lambda = diag(0.1, 0.2, 0.3, 0.4, 5, 6, 7, \ldots, 100)$. The right-hand side vector f is a normalized vector of all ones.

In Figure 3.3.1, we plot convergence curves for GMRES and GCRO-DR(24,4) applied to the system $A^{(1)} = f$. Note that the matrix $A^{(1)}$ is SPD. Except for the first cycle, GCRO-DR performs 20 matrix-vector multiplications in each cycle. GCRO-DR was asked to recycle four vectors in order to investigate its ability to pick up the four smallest eigenvalues

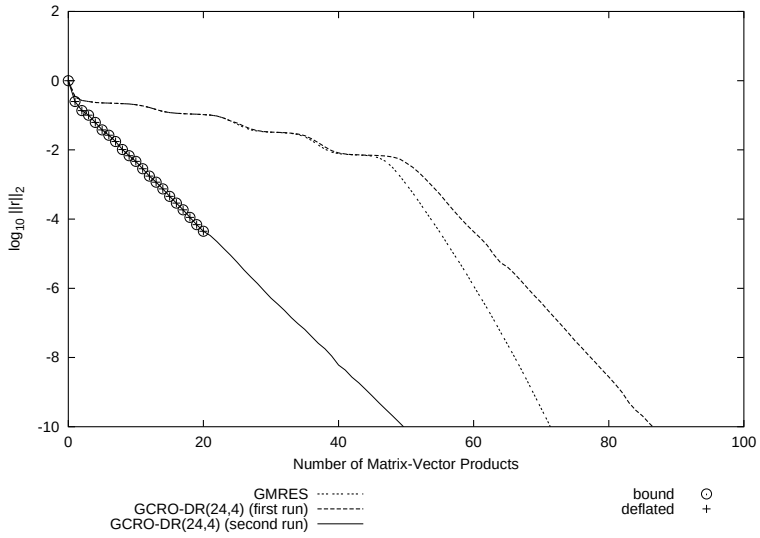


Figure 3.3.1: Example 3.3.1, $\kappa\left(S^{(1)}\right) = 1$. Number of matrix-vector multiplications vs. residual norm for various solvers. In the legend, "bound" represents the bound (3.3.8), and "deflated" represents the deflated problem (3.3.3). For the bound, Q_ℓ ($\ell = 4$) was selected to be the the span of the four eigenvectors corresponding to the four eigenvalues of smallest magnitude.

$0.1, 0.2, 0.3$, and 0.4. We see that the bound (3.3.8) is nearly identical to the actual convergence curve, and that the GCRO-DR curve appears to line up with the deflated problem (3.3.3). In Table 3.3.1 we plot some terms from the bound (3.3.8). When computing the bound for this case, the subspace $\mathcal{R}(Q_\ell)$ $(\ell = 4)$ was selected to be the span of the four eigenvectors corresponding to the four eigenvalues of smallest magnitude. In this case, GCRO-DR was successful in selecting and recycling an invariant subspace, and removing the effects of all components in that invariant subspace.

In Figure 3.3.2, we plot convergence curves for GMRES and GCRO-DR(24,4) applied to the system $A^{(2)} = f$. We see that the bound (3.3.8) is nearly identical to the actual convergence curve, and that the GCRO-DR curve appears to line up with the deflated bound (3.3.3). In Table 3.3.2 we plot some terms from the bound (3.3.8). When computing the bound for this case, the subspace $\mathcal{R}(Q_\ell)$ $(\ell = 4)$ was selected to be the span of the four eigenvectors corresponding to the four eigenvalues of smallest magnitude. Despite the

Table 3.3.1: *Example* 3.3.1. $\kappa\left(S^{(1)}\right) = 1$.

GCRO-DR denotes the residual norm produced by GCRO-DR at iteration j.

$\kappa(A^{(1)})$	γ	$\|P_Q\|_2$	δ
10^3	1.8129e-08	1.0000e+00	1.8129e-08
j	GCRO-DR (2^{nd} run)	Bound (3.3.8)	Equation (3.3.3)
1	2.5052e-01	2.5052e-01	2.5052e-01
2	1.3648e-01	1.3648e-01	1.3648e-01
3	1.0051e-01	1.0051e-01	1.0051e-01
4	6.1982e-02	6.1982e-02	6.1982e-02
5	3.7868e-02	3.7868e-02	3.7868e-02
6	2.6543e-02	2.6543e-02	2.6543e-02

slightly ill-conditioned eigenbasis, GCRO-DR was successful in selecting and recycling an invariant subspace, and removing the effects of that invariant subspace. This example shows that convergence is not strongly affected by nonnormality, so long as $\|P_Q\|_2$ is small.

In Figure 3.3.3, we plot convergence curves for GMRES and GCRO-DR(68,1) applied to the system $A^{(3)} = f$. With a smaller restart parameter, the method resolved no eigenvectors. We see that the bound (3.3.8) is not close to the actual convergence curve in this case. In Table 3.3.3 we plot some terms from the bound (3.3.8). When computing the bound for this case, the subspace $\mathcal{R}(Q_\ell)$ $(\ell = 1)$ was selected to be the span of the eigenvector corresponding to the eigenvalue of smallest magnitude. Note that the factor γ indicates that GCRO-DR was not sufficiently successful in removing this one-dimensional invariant subspace. Referring to Proposition 3.3.2, we note that although δ is not large, the value of $\|P_Q\|_2$ allows for the possibility that γ may not be small. Note that in Figure 3.3.3 the convergence curve for GCRO-DR(68,1) is initially below the bound (3.3.8), and the deflated problem (3.3.3). This suggests that a deflationary approach (e.g. problem (3.3.3)) is not ideal, especially in the case where $\|P_Q\|_2$ is large.

In Figure 3.3.4, we examine GCRO-DR(44,4) on the linear system $A^{(3)}x = f$. We let

Table 3.3.2: *Example* 3.3.1. $\kappa\left(S^{(2)}\right) = 10^3$.

GCRO-DR denotes the residual norm produced by GCRO-DR at iteration j.

$\kappa(A^{(2)})$	γ	$\|P_Q\|_2$	δ
5.9458e+04	2.0715e-07	2.0302e+00	1.3504e-007
j	GCRO-DR (2^{nd} run)	Bound (3.3.8)	Equation (3.3.3)
1	7.0565e-01	7.3202e-01	7.3202e-01
2	4.4612e-01	4.7169e-01	4.7169e-01
3	3.7762e-01	4.0096e-01	4.0095e-01
4	2.0057e-01	2.2508e-01	2.2508e-01
5	1.4790e-01	1.7091e-01	1.7091e-01
6	9.8155e-02	1.1478e-01	1.1478e-01

Table 3.3.3: *Example* 3.3.1. $\kappa\left(S^{(3)}\right) = 10^6$

GCRO-DR denotes the residual norm produced by GCRO-DR at iteration j.

$\kappa(A^{(3)})$	γ	$\|P_Q\|_2$	δ
4.9383e+010	7.4978e-003	1.6925e+003	4.4299e-006
j	GCRO-DR (2^{nd} run)	Bound (3.3.8)	Equation (3.3.3)
1	4.8673e-01	1.7456e+02	1.6838e+02
2	4.7134e-01	1.7087e+02	1.6489e+02
3	3.1344e-01	1.5139e+02	1.4628e+02
4	3.0455e-01	1.5026e+02	1.1065e+02
5	2.4534e-01	1.1451e+02	7.3275e+01
6	2.3129e-01	1.1451e+02	4.9519e+01

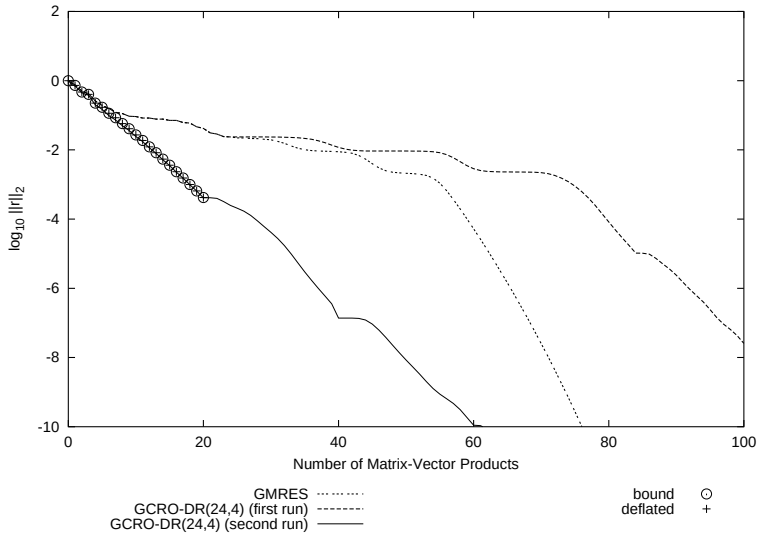


Figure 3.3.2: Example 3.3.1, $\kappa\left(S^{(2)}\right) = 10^3$. Number of matrix-vector multiplications vs. residual norm for various solvers. In the legend, "bound" represents the bound (3.3.8), and "deflated" represents the deflated problem (3.3.3). For the bound, Q_ℓ ($\ell = 4$) was selected to be the span of the four eigenvectors corresponding to the four eigenvalues of smallest magnitude.

GCRO-DR recycle the subspace it selected at the end of the first run, and compare with a different GCRO-DR process that recycles the invariant subspace spanned by the four eigenvectors corresponding to the four eigenvalues of smallest magnitude. Note that the subspace selected by GCRO-DR produces a smaller residual norm for almost the entire run. This means that the four vectors spanning an approximate invariant subspace were more useful for convergence than the eigenvectors themselves, again suggesting that invariant subspaces are not the optimal choice when selecting a subspace to recycle. In particular, the invariant subspace selected by GCRO-DR proved a better choice than an invariant subspace. We examine this notion further in Example 3.3.2.

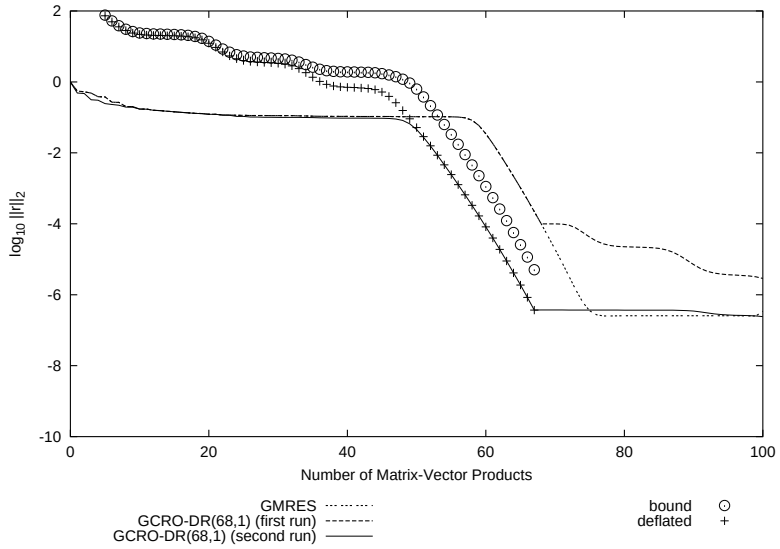


Figure 3.3.3: Example 3.3.1, $\kappa\left(S^{(3)}\right) = 10^6$. Number of matrix-vector multiplications vs. residual norm for various solvers. In the legend, "bound" represents the bound (3.3.8), and "deflated" represents the deflated problem (3.3.3). For the bound, Q_ℓ $(\ell = 1)$ was selected to be the span of the single eigenvector corresponding to the eigenvalue of smallest magnitude.

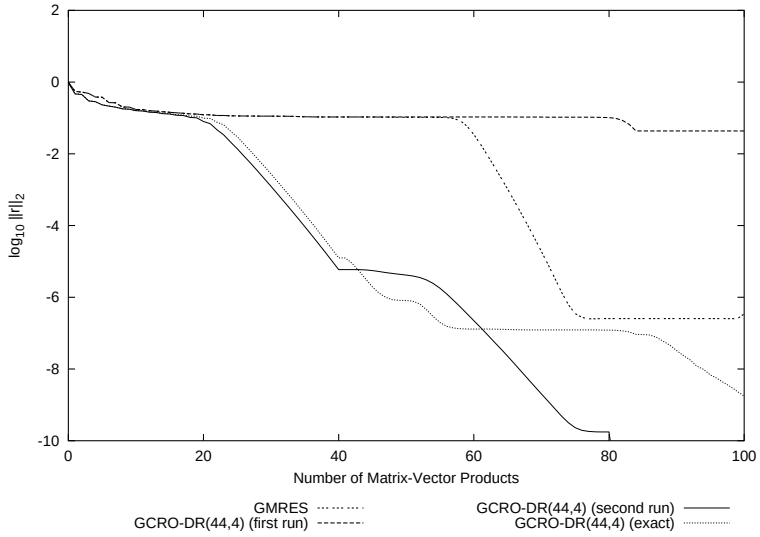


Figure 3.3.4: Example 3.3.1, $\kappa\left(S^{(3)}\right) = 10^6$. Number of matrix-vector multiplications vs. residual norm for various solvers. In this case, the invariant subspace corresponding to the four smallest eigenvalues was recycled. Note that recycling the exact invariant subspace produces worse results than the subspace selected by GCRO-DR.

EXAMPLE 3.3.2. 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

$$u(x,0) = u(0,y) = 0,$$
$$u(x,1) = u(1,y) = 1.$$

Central differences are used, and we set the mesh width to be $h = 1/26$ in both directions, which results in a 625×625 matrix. We consider the symmetric $c = 0$ case and the nonsymmetric $c = 25$ case.

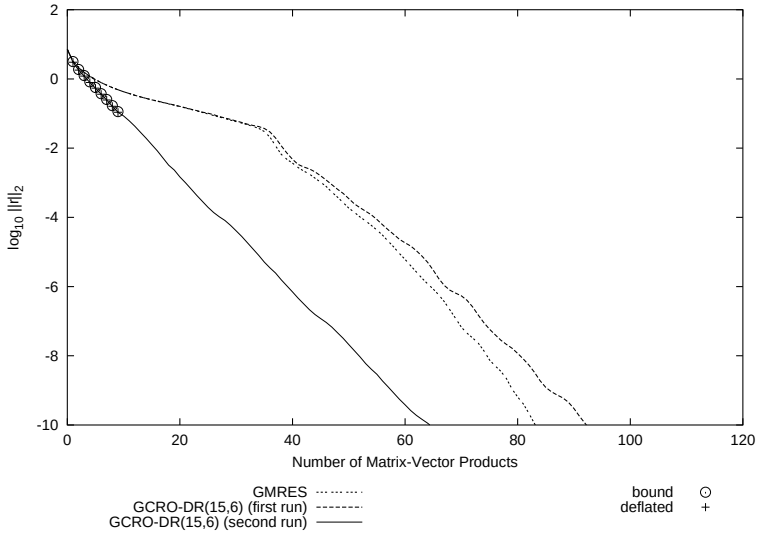


Figure 3.3.5: Example 3.3.2, $c = 0$ (Hermitian) case. Number of matrix-vector multiplications vs. residual norm for various solvers. In the legend, "bound" represents the bound (3.3.8), and "deflated" represents the deflated problem (3.3.3). For the bound, Q_ℓ $(\ell = 6)$ was selected to be the span of the six eigenvectors corresponding to the six eigenvalues of smallest magnitude. Note that the deflated bound lines up exactly with the GCRO-DR convergence curve.

Table 3.3.4: *Example* 3.3.2. $c = 0$. Eigenvalues, numbered from smallest magnitude, along with the inner product of the right-hand side and the eigenvector associated with each eigenvalue. Eigenvalues in italics correspond to eigenvectors selected by GCRO-DR at the end of its first run in Figure 3.3.8. The eigenvectors associated with eigenvalues 1,2,3,5,6,7 were used for the run "exact 1-6", and the eigenvectors associated with eigenvalues 8,9,10,11,14,15 were used for the run "exact 7-12".

Index	Component in RHS	Eigenvalue
1	*0.30657059601507*	*-0.02916450360779*
2	0.00086174007236	-0.07269861695179
3	*0.43039431848736*	*-0.07269861695179*
4	0.00000000000002	-0.11623273029580
5	0.00068793691415	-0.14454976643306
6	*0.70928923205265*	*-0.14454976643307*
7	0.00019721614732	-0.18808387977706
8	*0.14206341536415*	*-0.18808387977708*
9	0.01706190332871	-0.24367020049746
10	*0.83560318634045*	*-0.24367020049747*
11	*0.29769480656712*	*-0.25993502925835*
12	0.00000000000001	-0.28720431384147
13	0.00000000000003	-0.28720431384148
14	0.00337731928758	-0.35905546332275
15	0.27585021322494	-0.35905546332275

In Figure 3.3.5, we plot convergence curves for GMRES and GCRO-DR(15,6) applied to the $c = 0$ system. Note that this system is SPD. We see that the bound (3.3.8) is very close to the actual convergence curve, and that the GCRO-DR curve appears to line up with the deflated bound (3.3.3). When computing the bound for this case, the subspace $\mathcal{R}(Q_\ell)$ $(\ell = 6)$ was selected to be the span of the eigenvectors 1, 3, 6, 8, 10, and 11, where the eigenvectors have been numbered starting with the corresponding eigenvalue of smallest magnitude and moving away from the origin. Note that some of the eigenvectors correspond to repeated eigenvectors, and that the right-hand side vector does not have components in the direction of all eigenvectors. As with the previous Hermitian example, GCRO-DR was successful in selecting and recycling an invariant subspace, and removing from the right-hand side all components in that invariant subspace.

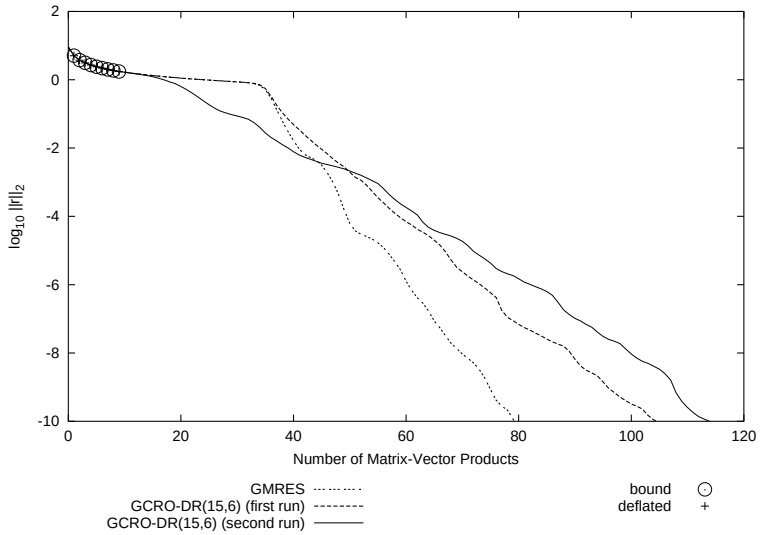


Figure 3.3.6: Example 3.3.2, $c = 25$ case. Number of matrix-vector multiplications vs. residual norm for various solvers. In the legend, "bound" represents the bound (3.3.8), and "deflated" represents the deflated problem (3.3.3). For the bound, Q_ℓ $(\ell = 2)$ was selected to be the span of the two eigenvectors corresponding to the two eigenvalues of smallest magnitude. The deflated problem (3.3.3) tracks nearly on top of the GCRO-DR curve. A subspace of dimension 6 was recycled, but only captured an invariant subspace of dimension 2. Note that the first run of GCRO-DR converges before the second run.

In Figure 3.3.6, we plot convergence curves for GMRES and GCRO-DR(15,6) applied to the $c = 25$ system. This system is not Hermitian, and the condition number of the eigenvector matrix is $3.0495e+05$. We see that the bound (3.3.8) is very close to the actual convergence curve, and that the GCRO-DR curve appears to line up with the deflated problem (3.3.3). When computing the bound for this case, the subspace $\mathcal{R}(Q_\ell)$ $(\ell = 2)$ was selected to be the span of the eigenvectors corresponding to the *two* eigenvalues of smallest magnitude. In this case,we see that GCRO-DR was successful in selecting and recycling an invariant subspace, and removing from the right-hand side all components in that invariant subspace. Note however that GCRO-DR recycled a subspace of dimension 6, but only captured an invariant subspace of dimension 2.

Of additional interest, we see in Figure 3.3.6 that the first run of GCRO-DR converges before the second run, even though the second run utilized the subspace recycled from the

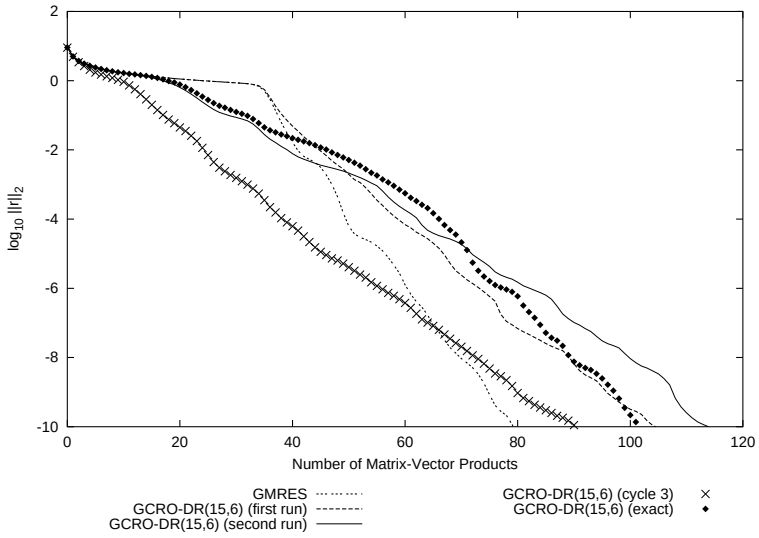


Figure 3.3.7: Example 3.3.2, $c = 25$ case. Number of matrix-vector multiplications vs. residual norm for various solvers. "exact" refers to a GCRO-DR process that started with the six eigenvectors from the six eigenvalues of smallest magnitude. "cycle 3" refers to a GCRO-DR process that starts with the subspace determined after the third cycle of the first run of GCRO-DR.

first run. Clearly, the recycled subspace was not useful for convergence.

This raises the question of how to select the "best" subspace to recycle, which we consider in Figure 3.3.7. Clearly, the invariant subspace corresponding to the six eigenvalues of smallest magnitude is not the best choice. The subspace selected at the end of the first GCRO-DR run is the worst choice shown. If we look at the first run of GCRO-DR, we see a sharp change in the convergence rate at the end of the third cycle (near iteration 33). The curve "cycle 3" shows the performance of GCRO-DR when recycling this subspace. Although this does not address the question of the optimal subspace to select, it suggests that recycling the subspace determined at the end of a GCRO-DR run is not always the best choice.

How does the choice of subspace affect convergence? Clearly, the actual convergence process is more complicated than simply removing invariant subspaces, especially those from the ends of the spectrum. Perhaps contrary to intuition, deflating away the eigenval-

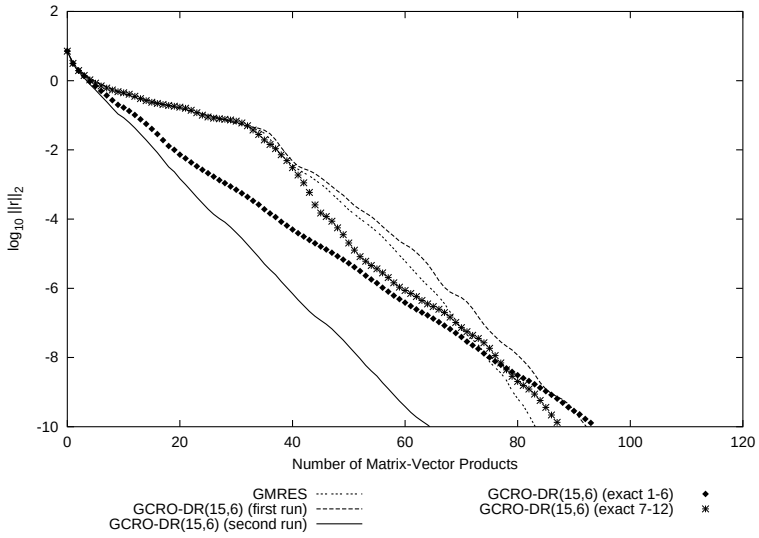


Figure 3.3.8: Example 3.3.2, $c = 0$ case. Number of matrix-vector multiplications vs. residual norm for various solvers. "exact 1-6" refers to a GCRO-DR process that started with the six eigenvectors from the six eigenvalues of smallest magnitude. "exact 7-12" is analogous. Note that although "exact 1-6" reduced the condition number of the problem, "exact 7-12" converged first.

ues closest to the origin is not always best. We believe that the convergence *process* is important. GCRO-DR always recycles the subspace corresponding to the harmonic Ritz vectors of smallest magnitude. If the GCRO-DR process starts with the k eigenvectors corresponding to the k eigenvalues of smallest magnitude, it will always recycle those vectors, and becomes identical to restarted GMRES on a deflated problem. If, instead, a subspace (such as the one recycled from cycle #3 in Figure 3.3.7) is kept, intermediate eigenvalues are removed, and the resulting spectrum may appear more clustered, effectively "preconditioning" the iteration. Later GCRO-DR iterations will then recycle and remove invariant subspaces corresponding to smaller eigenvalues.

Consider Figure 3.3.8, where we plot convergence curves for solvers applied to the $c = 0$ (Hermitian) problem, where the convergence is governed exclusively by the spectra. Here, we compare a GCRO-DR process started with the six eigenvectors corresponding to the six eigenvalues of smallest magnitude with a GCRO-DR process started with the

eigenvectors corresponding to the 7-12 eigenvalues of smallest magnitude. Eigenvectors orthogonal to the right-hand side were not included in the "exact" invariant subspaces. Surprisingly (or perhaps not) the latter of the two processes converges first. This may be due to clustering of the smaller magnitude eigenvalues, or to the fact that more of the right-hand side vector is contained in eigenvectors 7-12 than in eigenvectors 1-6, as shown in Table 3.3.4. More importantly, we see in Table 3.3.4 the eigenvectors and eigenvalues selected by GCRO-DR after its first run. Among the choices shown in Figure 3.3.8, this choice is clearly best. We can see from Table 3.3.4 that the eigenvectors selected correspond to the eigenvalues of smallest magnitude where the associated eigenvector is more strongly oriented with the right-hand side. As we can see from Table 3.3.4, some eigenvalues are repeated. For a repeated eigenvalue, a Krylov method only sees one eigenvector, which is the projection of the right-hand side onto the invariant subspace associated with the repeated eigenvalue. When GCRO-DR recycles an approximate eigenvector, it will select this single eigenvector. As such, although GCRO-DR only explicitly recycles k approximate eigenvectors, it may be effectively recycling many more than k eigenvectors. This benefit occurs only in linear systems with repeated eigenvalues. However, repeated eigenvalues frequently arise naturally in physical systems, as a consequence of symmetry.

EXAMPLE 3.3.3. As we can see from Example 3.3.2, the choice of the recycled subspace can seriously impact convergence. In this example, we consider a 100×100 real matrix A generated with a random eigenbasis, but only 10 distinct eigenvalues $1, 2, \ldots, 10$. Thus, GMRES will converge in at most 10 steps. The condition number of the matrix is approximately $5.16e5$, and the condition number of the eigenvector matrix is approximately $8.69e3$. The right-hand side vector is a random vector of unit norm. We consider two GCRO-DR processes. The first recycles the subspace generated from an initial run of GCRO-DR, and the second recycles a randomly generated subspace. For the second case, we suppose that there was a large perturbation from one matrix to the next in the sequence,

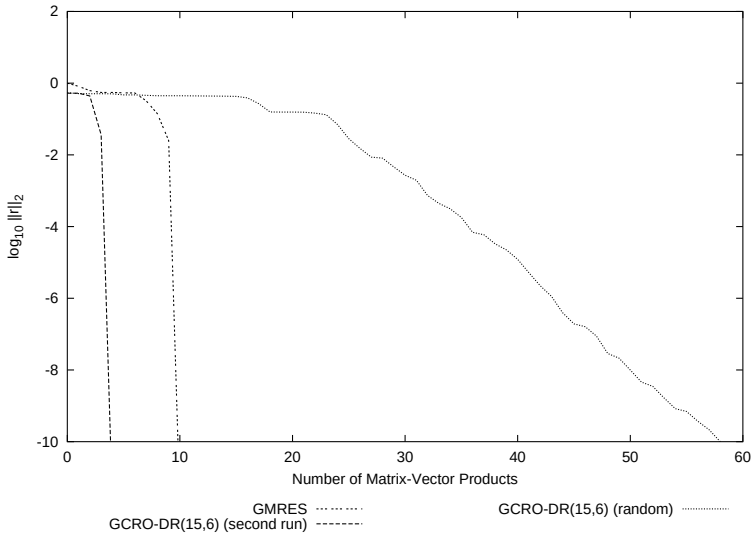


Figure 3.3.9: Example 3.3.3. Number of matrix-vector multiplications vs. residual norm for various solvers.

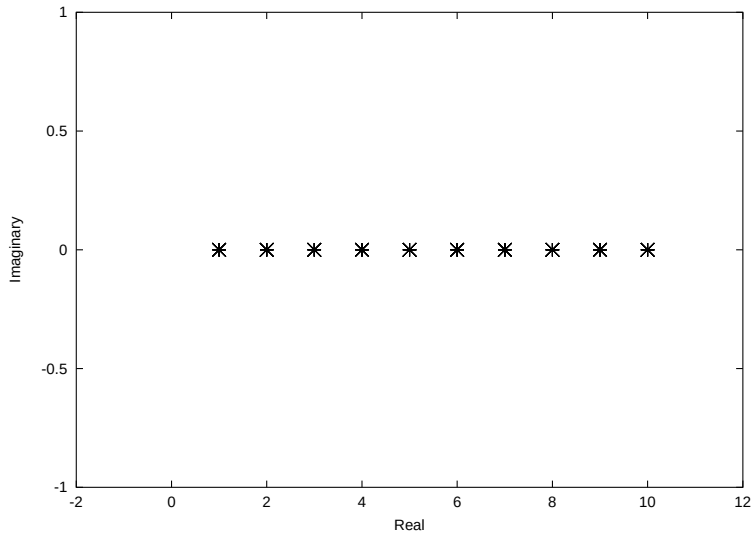


Figure 3.3.10: Example 3.3.3. Nonzero eigenvalues of $\left(I - C_1 C_1^H\right) A$, where C_1 determined by recycling subspace from first run.

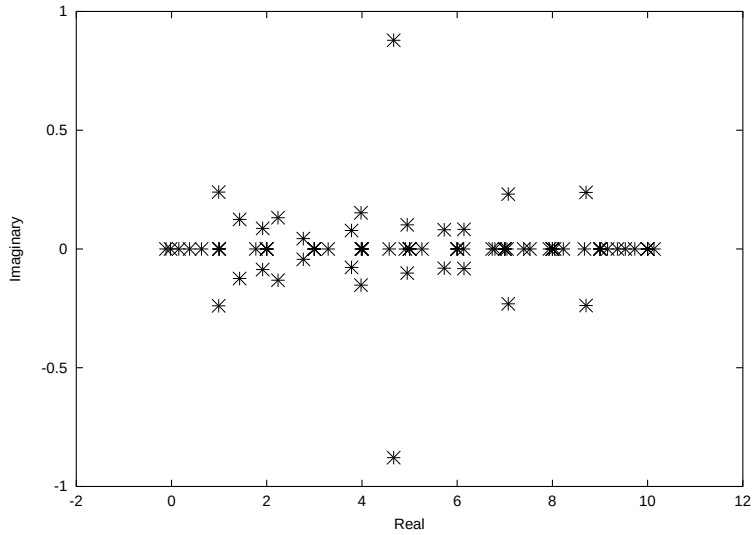


Figure 3.3.11: Example 3.3.3. Nonzero eigenvalues of $\left(I - C_2 C_2^H\right) A$, where C_2 random.

and that while the recycled subspace may have been a good approximation to an invariant subspace for the previous matrix, it is essentially random with respect to the next matrix. Let C_1 denote the first subspace, and C_2 the second (random) subspace. Figure 3.3.9 shows the convergence curve of the first and second GCRO-DR processes, as well as full GMRES. We see that the convergence is significantly worse when the random subspace is used, and that far more than 10 iterations are required. This behavior can be explained by examining Figures 3.3.10 and 3.3.11. Figure 3.3.10 shows the nonzero eigenvalues of $\left(I - C_1 C_1^H\right) A$, and Figure 3.3.11 shows the nonzero eigenvalues of $\left(I - C_2 C_2^H\right) A$. The random subspace scattered the eigenvalues, meaning that far more than 10 iterations may be required for convergence. As such, we see that a poorly chosen subspace can have catastrophic effects on convergence.

3.4 Concluding Remarks

We have presented an analytical model describing the convergence of deflation-based Krylov subspace recycling. The analysis shows that if the recycled subspace $\mathcal{R}(C_k)$ contains an

invariant subspace $\mathcal{R}(Q_\ell)$, convergence can be bounded above by problem (3.3.3). Experimental evidence supports this conclusion, but also shows that a deflationary approach is not optimal. Specifically, there exist better choices than simply trying to deflate the eigenvalues closest to the origin.

Chapter 4

KKT Preconditioners for FETI Methods: New Connections

Preconditioners for KKT (Karush-Kuhn-Tucker) linear systems have been studied extensively. The one-level finite element tearing and interconnecting (FETI) [16] method produces a linear system of this form. In this chapter, we show new connections between recently proposed KKT preconditioners and solvers and the one-level FETI method. These connections provide a new perspective on the analysis of FETI preconditioners by leveraging work for KKT systems. In particular, they provide a means of bounding the eigenvalues of preconditioned FETI systems, and thus the rate of convergence of an iterative solver. This theoretical framework gives a means to analyze the usefulness of improvements to FETI preconditioners.

The FETI method requires the solution of an expensive subproblem, in which a Schur complement matrix is factorized. Connections we will demonstrate allow us to extend the FETI method to allow for the use of an approximate Schur complement. This has several advantages, the first being reduced computational cost. When solving a sequence of FETI problems, we can amortize the cost of this subproblem by "recycling" the factorized Schur complement matrix for the next linear system, and using it as an approximation to the true Schur complement matrix for the next linear system. We can also bound the locations of the eigenvalues for a preconditioned FETI system using an inexact Schur complement, and

thus predict how convergence is affected by the inexact Schur complement.

4.1 Introduction

The one-level finite element tearing and interconnecting (FETI) method was one of the first domain decomposition methods to exhibit numerical scalability with respect to both the mesh and subdomain sizes [15], when equipped with an appropriate preconditioner. In section 4.2 we review the original FETI method and its traditional preconditioners, as described in [16]. We will find that the FETI method requires the solution of a KKT system, and that the solution to this KKT system is computed by forming and solving a reduced-size linear system. In section 4.3 we discuss a class of block-diagonal KKT preconditioners. In section 4.4 we show an equivalence between these block-diagonal preconditioners and FETI preconditioners. In connection with this class of block-diagonal KKT preconditioners, a reduced-size "related system" was proposed. In section 4.5 we outline this so-called "related system", and show equivalence between the related system and the reduced-size problem solved by the FETI method. In section 4.6 we show applications of these newly developed insights about the FETI method. These algebraic connections provide a new means by which to analyze existing FETI preconditioners, and suggest how to construct new ones. We bound the spectrum of the FETI dual interface problem. We also develop a FETI method that uses an approximate Schur complement, and bound the locations of the eigenvalues of the preconditioned system. We offer concluding remarks in 4.7.

4.2 Review of the One-Level FETI Method

The FETI method is a domain decomposition method that solves iteratively the linear system of equations arising from the finite element discretization of self-adjoint elliptic partial differential equations.

Suppose for a domain Ω that the associated linear system is $Ku = f$, where K is the global stiffness matrix, u is the unknown vector of displacements for each degree of freedom, and f is a vector of applied forces. The FETI method proceeds by cutting the domain into pieces, and then solving a local problem on each subdomain, with the requirement that the solution is continuous across subdomain boundaries. If Ω is "torn" into N_s nonoverlapping regions $\{\Omega^{(s)}\}_{s=1}^{s=N_s}$, then FETI replaces the global problem with N_s subproblems,

$$K^{(s)}u^{(s)} = f^{(s)} - {B^{(s)}}^T\lambda, \qquad s = 1,\ldots,N_s \tag{4.2.1a}$$

$$\sum_{s=1}^{N_s} B^{(s)}u^{(s)} = 0, \tag{4.2.1b}$$

where $K^{(s)}$ is the stiffness matrix, $u^{(s)}$ the displacement vector, and $f^{(s)}$ the prescribed force vector associated with the finite element discretization of the region $\Omega^{(s)}$, and $B^{(s)}$ is a signed boolean matrix that extracts and signs the interface components of a vector. Equation (4.2.1b) represents the set of constraints that require the subdomains $\Omega^{(s)}$ be continuous along their interface $\Gamma^{(s)}$. The vector of Lagrange multipliers λ represents the forces between the subdomains at their interface. Clearly, once λ has been determined, each of the N_s subproblems (4.2.1a) is now completely decoupled and can be solved in an embarrassingly parallel manner.

The "tearing" process often generates substructures which do not have enough prescribed displacements to locally eliminate rigid body modes. If, for example, the domain $\Omega^{(j)}$ does not have enough prescribed boundary conditions, then the local stiffness matrix $K^{(j)}$ is semi-definite, and special attention must be given to the equation

$$K^{(j)}u^{(j)} \quad = f^{(j)} - {B^{(j)}}^T\lambda. \tag{4.2.2}$$

If this singular system is consistent, the general solution of equation (4.2.2) is given by

$$u^{(j)} = K^{(j)^{\dagger}} \left(f^{(j)} - B^{(j)^T} \lambda \right) + R^{(j)} \alpha^{(j)}, \tag{4.2.3}$$

where $K^{(j)^{\dagger}}$ is the pseudoinverse of $K^{(j)}$, $R^{(j)}$ is a rectangular matrix whose columns form a basis for the null space of $K^{(j)}$, and $\alpha^{(j)}$ specifies the contribution from the null space $R^{(j)}$ to the local solution $u^{(j)}$. Physically, $R^{(j)}$ represents the rigid body (zero energy) modes of $\Omega^{(j)}$, and $\alpha^{(j)}$ specifies a particular linear combination of these. Note that if $K^{(j)}$ is singular, for (4.2.2) to have a solution, $f^{(j)} - B^{(j)^T} \lambda$ must be in $\mathcal{R}\left(K^{(j)}\right)$. This requires that

$$R^{(j)^T} \left(f^{(j)} - B^{(j)^T} \lambda \right) = 0. \tag{4.2.4}$$

4.2.1 The FETI Dual Interface Problem

If we substitute (4.2.3) into (4.2.1b) and exploit (4.2.4), the equations (4.2.1a-4.2.1b) can be formulated equivalently as

$$\begin{bmatrix} F & -G \\ -G^T & 0 \end{bmatrix} \begin{bmatrix} \lambda \\ \alpha \end{bmatrix} = \begin{bmatrix} d \\ -e \end{bmatrix}, \tag{4.2.5}$$

where the matrix $F \in \mathbb{R}^{n \times n}$ is symmetric positive semi-definite (SPSD), and $G \in \mathbb{R}^{n \times m}$ is full rank, where $n \geq m$. System (4.2.5) is called the *dual interface problem* because λ is the

dual variable to the primal variables $u^{(s)}$. The submatrices in (4.2.5) are defined as

$$
\begin{aligned}
F &= \sum_{s=1}^{N_s} B^{(s)} K^{(s)^\dagger} B^{(s)^T}, \\
d &= \sum_{s=1}^{N_s} B^{(s)} K^{(s)^\dagger} f^{(s)}, \\
G_I &= \left[G_I^{(1)} \dots G_I^{(N_f)} \right] = \left[B^{(1)} R^{(1)} \dots B^{(N_f)} R^{(N_f)} \right], \\
\alpha &= \left[\alpha^{(1)} \dots \alpha^{(N_f)} \right], \\
e &= \left[f^{(1)^T} R^{(1)} \dots f^{(N_f)^T} R^{(N_f)} \right],
\end{aligned}
$$

where N_f denotes the number of floating subdomains. We refer to the second block equation of (4.2.5), $G^T \lambda = e$, as the constraint equations.

In practice, the matrix F is never explicitly assembled. Instead, the dual interface problem is solved iteratively with the preconditioned conjugate gradient (PCPG) algorithm, discussed in section 4.2.2, which requires only multiplication by the matrix F.

4.2.2 Iterative Solution of the Dual Interface Problem

In the FETI method, the indefinite dual interface problem (4.2.5) is transformed into a smaller positive semidefinite system by first satisfying the constraint equations. This positive semidefinite system is solved iteratively, and the constraint is explicitly maintained by projection. To satisfy the constraint equations, we must choose a $\lambda^{(0)}$ such that $G^T \lambda^{(0)} = e$. For this, we select

$$
\lambda^{(0)} = Q G_I \left(G_I^T Q G_I \right)^{-1} e. \tag{4.2.6}
$$

This choice effectively decomposes the solution vector λ as

$$\lambda = \lambda^{(0)} + \Delta\lambda, \tag{4.2.7}$$

where $\Delta\lambda \in \ker\left(G^T\right)$. The most direct way of doing this is to introduce the projector

$$P(Q) = I - QG_I\left(G_I^T QG_I\right)^{-1} G_I^T \tag{4.2.8}$$

onto $\ker\left(G^T\right)$, where Q is any matrix such that $(G_I^T QG_I)^{-1}$ exists and is SPD. We will discuss choices for Q below. Letting $\Delta\lambda = P(Q)\xi$ for some ξ, the first block of equations in (4.2.5) may be written as

$$FP(Q)\xi = d - F\lambda^{(0)} + G\alpha. \tag{4.2.9}$$

Left multiplication of the system (4.2.9) by the projector $P(Q)^T$ decouples α, restores symmetry, and transforms the FETI dual interface problem into the *projected interface problem*

$$\left(P(Q)^T FP(Q)\right)\xi = P(Q)^T\left(d - F\lambda^{(0)}\right). \tag{4.2.10}$$

Once ξ has been determined, we can express the solution λ as $\lambda = \lambda^{(0)} + P(Q)\xi$. We then solve for the rigid body mode coefficients

$$\alpha = -\left(G_I^T QG_I\right)^{-1} G_I^T Q\left(d - F\lambda\right), \tag{4.2.11}$$

and finally for the subdomain solutions (4.2.3), which can be computed concurrently for each subdomain.

In practice, the projected interface problem (4.2.10) is solved using the Preconditioned Conjugate Projected Gradient algorithm (PCPG) as shown in Algorithm 4.1, where M^{-1}

denotes a particular choice of preconditioner (discussed in section 4.2.3). Note that Algorithm 4.1 has been written to iterate directly on $\lambda^{(k)} = \lambda^{(0)} + P(Q)\xi^{(k)}$, rather than $\xi^{(k)}$, and $\lambda^{(0)}$ denotes the initial guess for the iterative method in this case.

Algorithm 4.1: **Preconditioned Conjugate Projected Gradient (PCPG)**

1: $\lambda^{(0)} = QG_I \left(G_I^T Q G_I\right)^{-1} e$

2: $w^{(0)} = P^T \left(d - F\lambda^{(0)}\right)$

3: **for** $k = 0, 1, \ldots$ **do**

4: $\quad y^{(k)} = PM^{-1}w^{(k)}$

5: $\quad p^{(k)} = y^{(k)} - \sum_{i=0}^{k-1} \frac{y^{(k)^T} F p^{(i)}}{p^{(i)^T} F p^{(i)}} p^{(i)}$

6: $\quad \eta^{(k)} = \frac{p^{(k)^T} p^{(k)}}{p^{(k)^T} F p^{(k)}}$

7: $\quad \lambda^{(k+1)} = \lambda^{(k)} + \eta^{(k)} p^{(k)}$

8: $\quad w^{(k+1)} = w^{(k)} - \eta^{(k)} P^T F p^{(k)}$

9: **end for**

Application of $P(Q)$ requires the solution of the coarse space problem

$$\left(G_I^T Q G_I\right) \mu = \eta,$$

which couples all the subdomain equations, propagates error globally, and accelerates convergence [26]. There are several possible choices for Q in $P(Q)$. The simplest is $Q = I$, which is a computationally efficient choice for homogeneous problems [15]. For heterogeneous problems, it was proposed in [16] to set Q equal to the FETI lumped or Dirichlet preconditioners, which are discussed in section 4.2.3.

Note that PCPG keeps a full recurrence. From numerical experiments, it has been established that while most of the eigenvalues of F cluster near zero, a handful accumulate to a larger value [38]. This distribution of eigenvalues is known to cause PCPG to lose orthogonality, which slows convergence.

The usual FETI implementation generates redundant constraints at crosspoints of the finite element mesh (points belonging to three or more subdomains). As a result, F is semidefinite in these cases [15]. However, this condition can easily be rectified by eliminating the redundant constraints. This is not done in practice because the interface problem $P^T F P$ is nonsingular over the space over which we seek a solution [50]. Without loss of generality, we will assume that F is SPD for the remainder of this chapter.

For second order elasticity problems, the condition number of the dual interface problem grows asymptotically as

$$\kappa = O\left(1 + \log^2\left(\frac{H}{h}\right)\right) \tag{4.2.12}$$

when the Dirichlet preconditioner (discussed in section 4.2.3) is applied, where H denotes the subdomain size and h the mesh size [15]. This result details the numerical scalability of the FETI method for these problem classes. That is, if the mesh and subdomain sizes are refined so that H/h remains constant, the number of FETI iterations required to solve the problem is bounded by a constant.

4.2.3 Classical Preconditioners

We seek a matrix M^{-1} that approximates the inverse of F over the nullspace of G^T. Since F is not explicitly assembled, we would like to compute the preconditioner without having F explicitly available. Two commonly used preconditioners in the FETI literature that meet these requirements are the *lumped* and *Dirichlet* preconditioners [16].

Assume that the matrix $K^{(s)}$ is partitioned such that its internal degrees of freedom (DOFs) are numbered first. We will denote these by the subscript i, and the boundary

DOFs by the subscript b. We can then write

$$K^{(s)} = \begin{bmatrix} K_{ii}^{(s)} & K_{ib}^{(s)} \\ K_{bi}^{(s)} & K_{bb}^{(s)} \end{bmatrix}.$$

The lumped preconditioner derives its name because, from a mechanical viewpoint, it corresponds to finding a set of "lumped" interface forces that can reproduce the displacement jumps at the substructure interfaces when only the interface DOFs are allowed to displace. Since F is represented as the sum of matrices

$$F = \sum_{s=1}^{N_s} B^{(s)} K^{(s)^\dagger} B^{(s)^T},$$

the lumped preconditioner [1]can be expressed as

$$\begin{aligned} \left(M^L\right)^{-1} &= \sum_{s=1}^{N_s} W^{(s)} B^{(s)} K^{(s)} B^{(s)^T} W^{(s)} \\ &= \sum_{s=1}^{N_s} W^{(s)} B^{(s)} \begin{bmatrix} 0 & 0 \\ 0 & K_{bb}^{(s)} \end{bmatrix} B^{(s)^T} W^{(s)}, \end{aligned}$$

where $W^{(s)}$ is a diagonal "topological scaling" matrix. In the homogeneous case, $W^{(s)}$ stores the inverse of the multiplicity of the corresponding interface DOF. For structurally heterogeneous models, entries on the diagonal of $W^{(s)}$ are adjusted accordingly to account for varying material properties [37].

The Dirichlet preconditioner is based on a further mechanical interpretation of the FETI

[1]The use of $^{-1}$ indicates that $\left(M^L\right)^{-1}$ and $\left(M^L\right)^{-1}$ should be viewed as preconditioners, and does not imply that M^L or M^D correspond to invertible matrices, although both preconditioners are nonsingular over the space over which they are applied.

algorithm and can be expressed as

$$\begin{aligned}\left(M^D\right)^{-1} &= \sum_{s=1}^{N_s} W^{(s)} B^{(s)} \begin{bmatrix} 0 & 0 \\ 0 & K_{bb}^{(s)} - K_{ib}^{(s)^T} K_{ii}^{(s)^{-1}} K_{ib}^{(s)} \end{bmatrix} B^{(s)^T} W^{(s)} \\ &= \sum_{s=1}^{N_s} W^{(s)} B^{(s)} \begin{bmatrix} 0 & 0 \\ 0 & S^{(s)} \end{bmatrix} B^{(s)^T} W^{(s)},\end{aligned}$$

where $S^{(s)} = K_{bb}^{(s)} - K_{ib}^{(s)^T} K_{ii}^{(s)^{-1}} K_{ib}^{(s)}$ is the Schur complement of $K^{(s)}$. Note that the effect of multiplication by $(M^D)^{-1}$ can be achieved without the formation of the Schur complement matrix.

The Dirichlet preconditioner is superior to the lumped preconditioner, and is considered to be mathematically optimal. The Dirichlet preconditioner is, however, more expensive than the lumped preconditioner. Because of this added cost, the lumped preconditioner can be computationally more efficient [16].

4.3 KKT Preconditioners

The FETI dual interface problem (4.2.5) is a KKT system. Rather than focusing on mechanical intuition to develop FETI preconditioners, let us approach the problem from a purely algebraic perspective by leveraging existing research on KKT preconditioners and solvers.

In [13], the KKT preconditioner

$$\begin{bmatrix} F^{-1} & 0 \\ 0 & \left(G^T F^{-1} G\right)^{-1} \end{bmatrix} \tag{4.3.1}$$

was proposed for nonsingular F. The nonsingular preconditioned system can be shown to

have at most three distinct eigenvalues, meaning that any Krylov method will converge in at most three iterations. Unfortunately, this preconditioner is far too expensive to be practical.

In section 4.4, we instead consider a symmetric version of a related block-diagonal preconditioner derived from (4.3.1) and introduced in [7]. This preconditioner takes the form

$$\begin{bmatrix} D^{-1} & 0 \\ 0 & \left(G^T D^{-1} G\right)^{-1} \end{bmatrix}, \tag{4.3.2}$$

where $F = D - E$. The matrix D is chosen so that it is easily invertible. For the present problem, we will choose (4.3.2) to be symmetric positive definite. This preconditioner can be regarded as an extension of the preconditioner (4.3.1), which chooses $D = F$.

In [7] it is observed that after preconditioning a KKT system by (4.3.2), a smaller "related system" can be derived. This related system can be viewed as a generalized form of the projected interface problem (4.2.10). We discuss the related system further in section 4.5.

4.4 Block-Diagonal Preconditioners

Here we consider the application of the block-diagonal preconditioner (4.3.2) to the dual interface problem (4.2.5). We show that the resulting reduced-size projected interface problem is equivalent to a preconditioned projected interface problem in the original FETI method. In particular, this means that all FETI preconditioners can be regarded as splittings of the $(1,1)$ block of (4.2.5).

4.4.1 Applying the Preconditioner

It is generally preferable to preserve symmetry, so we factor the preconditioner (4.3.2) and apply it to the interface problem (4.2.5) symmetrically. Let us choose a splitting $F = D - E$ and compute the Cholesky factorization of the preconditioner, giving

$$\begin{bmatrix} D^{-1} & 0 \\ 0 & \left(G^T D^{-1} G\right)^{-1} \end{bmatrix} = \begin{bmatrix} L_D^T & 0 \\ 0 & L_G^T \end{bmatrix} \begin{bmatrix} L_D & 0 \\ 0 & L_G \end{bmatrix}, \tag{4.4.1}$$

where we have assumed that D is SPD. Since G is guaranteed to be full rank [50], $G^T D^{-1} G$ is SPD if D is SPD. Symmetrically preconditioning the interface problem (4.2.5) gives

$$\begin{bmatrix} L_D F L_D^T & -L_D G L_G^T \\ -L_G G^T L_D^T & 0 \end{bmatrix} \begin{bmatrix} L_D^{-T} \lambda \\ L_G^{-T} \alpha \end{bmatrix} = \begin{bmatrix} L_D d \\ -L_G e \end{bmatrix},$$

which we rewrite as

$$\begin{bmatrix} \widetilde{F} & -\widetilde{G} \\ -\widetilde{G}^T & 0 \end{bmatrix} \begin{bmatrix} \widetilde{\lambda} \\ \widetilde{\alpha} \end{bmatrix} = \begin{bmatrix} \widetilde{d} \\ -\widetilde{e} \end{bmatrix}. \tag{4.4.2}$$

In (4.4.2), $\widetilde{G}$ has the useful property

$$\widetilde{G}^T \widetilde{G} = L_G G^T \left(L_D^T L_D\right) G L_G^T = L_G \left(G^T D^{-1} G\right) L_G^T = I,$$

and therefore $\left(\widetilde{G}\widetilde{G}^T\right)$ is an orthogonal projector.

At this point, we can now apply PCPG to the preconditioned system (4.4.2). In this case, the modified FETI projector arising from (4.4.2) can be written as

$$\widetilde{P} = I - \widetilde{G}\left(\widetilde{G}^T \widetilde{G}\right)^{-1} \widetilde{G}^T = I - \widetilde{G}\widetilde{G}^T. \tag{4.4.3}$$

Note that the projector $\widetilde{P}$ is always symmetric. Further, the coarse space problem $(G^T G)\mu = \eta$ that must be solved twice at each PCPG iteration has been implicitly handled by the block-diagonal preconditioner.

4.4.2 Block-Diagonal and FETI Preconditioners

We consider the solution of (4.4.2) by PCPG. We show that application of the block-diagonal preconditioner (4.3.2) with PCPG is equivalent to any FETI preconditioner.

For the projected interface problem (4.2.10) we will assume that the chosen FETI preconditioner is Q, and that the associated projector is defined as $P(Q)$. We will assume that the preconditioner Q is applied symmetrically. In the following, we will make the assumption that $Q = D^{-1}$. We represent the Cholesky factorization of Q as $Q = D^{-1} = L_D^T L_D$. The preconditioned projected interface problem can be expressed as

$$\left(L_D P(Q)^T F P(Q) L_D^T\right)\left(L_D^{-T}\xi\right) = L_D P(Q)^T\left(d - F\lambda^{(0)}\right). \tag{4.4.4}$$

The corresponding block-diagonally preconditioned projected interface problem is

$$\left(\widetilde{P}^T \widetilde{F} \widetilde{P}\right)\widetilde{\xi} = \widetilde{P}^T\left(\widetilde{d} - \widetilde{F}\widetilde{\lambda}^{(0)}\right), \tag{4.4.5}$$

where $\widetilde{\lambda}^{(0)} = \widetilde{G}\widetilde{e} = L_D^{-T}\lambda^{(0)}$ was defined in (4.4.2) and (4.2.6). We show below that the matrices and right-hand sides in equations (4.4.4) and (4.4.5) are identical. In this case, we have that $\widetilde{\xi} = L_D^{-T}\xi$. If we have that $\lambda = \lambda^{(0)} + P(Q)\xi$ and that $\widetilde{\lambda} = \widetilde{\lambda}^{(0)} + \widetilde{P}(Q)\widetilde{\xi}$, then $\widetilde{\lambda} = L_D^{-T}\lambda$, in agreement with the relation given in (4.4.2). We have thus shown that when preconditioning the dual interface problem (4.2.5) and solving the preconditioned system with PCPG, the choice of the splitting D^{-1} is equivalent to selection of any FETI preconditioner, such as those in section 4.2.3. This means that every FETI preconditioner can be viewed as a splitting of F, and any of the large body of literature of matrix splittings

[28] can now be applied to FETI preconditioners. This allows an algebraic, rather than mechanical, means of constructing new FETI preconditioners.

It remains to show equality of the matrices and right-hand sides in (4.4.4) and (4.4.5). We consider the matrices first. We can rewrite the matrix in (4.4.5) as

$$\begin{aligned}
\widetilde{P}^T\widetilde{F}\widetilde{P} &= \left(I-\widetilde{G}\widetilde{G}^T\right)\widetilde{F}\left(I-\widetilde{G}\widetilde{G}^T\right) \\
&= L_D\left(I-G\left(G^TQG\right)^{-1}G^TQ\right)F\left(I-QG\left(G^TQG\right)^{-1}G^T\right)L_D^T \\
&= L_DP(Q)^TFP(Q)L_D^T,
\end{aligned}$$

which is precisely the matrix in (4.4.4).

Similarly, We can rewrite the right-hand side in (4.4.5) as

$$\begin{aligned}
\widetilde{P}^T\left(\widetilde{d}-\widetilde{F}\widetilde{\lambda}^{(0)}\right) &= \left(I-\widetilde{G}\widetilde{G}^T\right)\left(\widetilde{d}-\widetilde{F}\widetilde{\lambda}^{(0)}\right) \\
&= L_D\left(I-G\left(G^TQG\right)^{-1}G^TQ\right)\left(d-F\lambda^{(0)}\right) \\
&= L_DP(Q)^T\left(d-F\lambda^{(0)}\right),
\end{aligned}$$

which is precisely the right-hand side in (4.4.4).

4.5 FETI and the Related System

Rather than applying PCPG to the preconditioned system (4.4.2), we can solve the so-called "related system" developed in [7], which we describe here. We first rewrite the matrix in (4.4.2) as

$$\mathbf{B}(\widetilde{S}) = \begin{bmatrix} I-\widetilde{S} & -\widetilde{G} \\ -\widetilde{G}^T & 0 \end{bmatrix},$$

where

$$\widetilde{S} \equiv I - \widetilde{F} = I - L_D F L_D^T. \tag{4.5.1}$$

We note that

$$\mathbf{B}(\widetilde{S}) = \mathbf{B}(0) - \begin{bmatrix} \widetilde{S} & 0 \\ 0 & 0 \end{bmatrix},$$

and that the explicit form for the inverse of $\mathbf{B}(0)$ is

$$\mathbf{B}(0)^{-1} = \begin{bmatrix} I - \widetilde{G}\widetilde{G}^T & -\widetilde{G} \\ -\widetilde{G}^T & -I \end{bmatrix} = \begin{bmatrix} \widetilde{P} & -\widetilde{G} \\ -\widetilde{G}^T & -I \end{bmatrix}.$$

We may then rewrite the system (4.4.2) as

$$\mathbf{B}(0) \begin{bmatrix} \widetilde{\lambda} \\ \widetilde{\alpha} \end{bmatrix} = \begin{bmatrix} \widetilde{S} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \widetilde{\lambda} \\ \widetilde{\alpha} \end{bmatrix} + \begin{bmatrix} \widetilde{d} \\ -\widetilde{e} \end{bmatrix},$$

and multiply through by $\mathbf{B}(0)^{-1}$ to get the fixed-point iteration

$$\begin{aligned} \begin{bmatrix} \widetilde{\lambda}^{(k+1)} \\ \widetilde{\alpha}^{(k+1)} \end{bmatrix} &= \begin{bmatrix} \widetilde{P}\widetilde{S}\widetilde{\lambda}^{(k)} \\ -\widetilde{G}^T\widetilde{S}\widetilde{\lambda}^{(k)} \end{bmatrix} + \begin{bmatrix} \widetilde{P}\widetilde{d} + \widetilde{G}\widetilde{e} \\ -\widetilde{G}^T\widetilde{d} + \widetilde{e} \end{bmatrix} \\ &= \begin{bmatrix} \widetilde{P}\widetilde{S}\widetilde{\lambda}^{(k)} \\ -\widetilde{G}^T\widetilde{S}\widetilde{\lambda}^{(k)} \end{bmatrix} + \begin{bmatrix} \widetilde{f} \\ \widetilde{g} \end{bmatrix}. \end{aligned} \tag{4.5.2}$$

We note that $\widetilde{\alpha}$ depends only on $\widetilde{\lambda}$, so we may compute $\widetilde{\lambda}$ first, then compute $\widetilde{\alpha}$ afterwards, just as in PCPG. Writing out the update for $\widetilde{\lambda}^{(k+1)}$ gives

$$\widetilde{\lambda}^{(k+1)} = \widetilde{P}\,\widetilde{S}\,\widetilde{\lambda}^{(k)} + \widetilde{f}. \tag{4.5.3}$$

The fixed-point $\widetilde{\lambda}$ of this system satisfies $\widetilde{\lambda} = \widetilde{P}\,\widetilde{S}\,\widetilde{\lambda} + \widetilde{f}$, which we rewrite as

$$\left(I - \widetilde{P}\widetilde{S}\right)\widetilde{\lambda} = \widetilde{f} \tag{4.5.4}$$

to produce the *related system*. Following the discussion in [7], we note that each fixed-point iterate (4.5.3) corresponds to a $\lambda^{(k+1)}$ that satisfies the original constraint equation $G^T\lambda^{(k+1)} = e$. Further, if a Krylov subspace method is used to solve (4.5.4) where the initial guess $\widetilde{\lambda}^{(0)}$ satisfies the constraint equations, it can be shown that the constraint equations will be satisfied at every iteration. It is recommended in [7] to apply one fixed-point iteration to develop an iterate that satisfies the constraint equations, and then use that iterate as an initial guess for a Krylov method. In the following, we will instead assume that (4.5.4) is solved with a Krylov method using $\widetilde{\lambda}^{(0)}$ as an initial guess. For this case, we solve the related system

$$\left(I - \widetilde{P}\widetilde{S}\right)\widetilde{\xi} = \widetilde{f} - \left(I - \widetilde{P}\widetilde{S}\right)\widetilde{\lambda}^{(0)}. \tag{4.5.5}$$

Next, we compare the related system (4.5.5) to the block-diagonally preconditioned projected interface problem (4.4.5), and show them to be equivalent.

4.5.1 The Related System

Here, we compare the two linear systems (4.5.5) and (4.4.5). Recall that we have already shown equivalence between (4.4.5) and the original FETI method using $Q = D^{-1}$. Tran-

sitively, this implies the related system is also equivalent. First, we compare the matrices, then the right-hand sides. Recall the definition of $\widetilde{S}$ in (4.5.1), and that the projector $\widetilde{P}$ is symmetric. The related system matrix $I - \widetilde{P}\widetilde{S}$ is nonsymmetric, which seems counterproductive, as the original system we are trying to solve is symmetric. However, over the space over which the related system matrix is applied, it is the same as the FETI projected interface operator, and therefore also symmetric. If we start with a consistent initial guess, then all iterates are in $\ker\left(\widetilde{G}^T\right)$. This is equivalent to right-multiplication by $\widetilde{P}$, as a projector applied to its own range is the identity. This produces

$$\begin{aligned}\left(I - \widetilde{P}\widetilde{S}\right)\widetilde{P} &= \widetilde{P} - \widetilde{P}\left(I - \widetilde{F}\right)\widetilde{P} \\ &= \widetilde{P}^T\widetilde{F}\widetilde{P},\end{aligned}$$

which is precisely the matrix in (4.4.5). However, note that $I - \widetilde{P}\widetilde{S}$ is nonsingular, while $\widetilde{P}^T\widetilde{F}\widetilde{P}$ is nonsingular only over the space $\ker\left(\widetilde{G}^T\right)$.

Now, we compare the right-hand side vectors. In the related system (4.5.5) we have

$$\begin{aligned}\widetilde{f} - \left(I - \widetilde{P}\,\widetilde{S}\right)\widetilde{\lambda}^{(0)} &= \widetilde{P}\widetilde{d} + \widetilde{G}\widetilde{e} - \left(I - \widetilde{P}\,\widetilde{S}\right)\widetilde{\lambda}^{(0)} \\ &= \widetilde{P}\widetilde{d} + \widetilde{P}\left(I - \widetilde{F}\right)\widetilde{\lambda}^{(0)} \\ &= \widetilde{P}^T\left(\widetilde{d} - \widetilde{F}\,\widetilde{\lambda}^{(0)}\right),\end{aligned}$$

which is precisely the right-hand side in (4.4.5).

4.5.2 Computing $\widetilde{\alpha}$

Here, we consider the computation of $\widetilde{\alpha}$ after $\widetilde{\lambda}$ has been determined. In the block-diagonally preconditioned PCPG algorithm, we compute $\widetilde{\alpha}$ as

$$\begin{aligned}\widetilde{\alpha} &= -\left(\widetilde{G}^T\widetilde{G}\right)^{-1}\widetilde{G}^T\left(\widetilde{d}-\widetilde{F}\,\widetilde{\lambda}\right)\\ &= -\widetilde{G}^T\left(\widetilde{d}-\widetilde{F}\,\widetilde{\lambda}\right).\end{aligned}$$

In the method of [7], we compute $\widetilde{\alpha}$ using the fixed point iteration (4.5.2), which produces

$$\begin{aligned}\widetilde{\alpha} &= -\widetilde{G}^T\widetilde{S}\,\widetilde{\lambda}+\widetilde{g}\\ &= -\widetilde{G}^T\left(I-\widetilde{F}\right)\widetilde{\lambda}-\widetilde{G}^T\widetilde{d}+\widetilde{e}\\ &= -\widetilde{G}^T\left(\widetilde{d}-\widetilde{F}\,\widetilde{\lambda}\right),\end{aligned}$$

exactly the expression found above.

We have shown algebraic equivalence of the related system (4.5.5) and the block-diagonally preconditioned projected interface problem (4.4.5). Through section 4.4.2 we also have the equivalence of the related system (4.5.5) and the projected interface problem (4.4.4) preconditioned with any FETI preconditioner, so long as $Q = D^{-1}$. This allows an alternate approach to constructing and analyzing FETI preconditioners. Further, existing KKT analysis may now be immediately applied to FETI systems. We show some consequences of these equivalences in section 4.6.

4.6 Results from Equivalences

We begin by bounding the eigenvalues of the related system in a cluster about one. Let λ_R denote an eigenvalue of the related system (4.5.5). If ζ is an eigenvector of (4.5.5), then

$\left(I - \widetilde{P}\,\widetilde{S}\right)\zeta = \lambda_R \zeta$, and it follows that

$$|1 - \lambda_R|_2 \leq \|\widetilde{S}\|_2,$$

because $\widetilde{P}$ is an orthogonal projector. To the extent that the FETI preconditioner becomes an exact inverse, $\|\widetilde{S}\|_2$ goes to zero.

The block-diagonal preconditioner (4.3.2) requires the inverse of the Schur complement matrix (coarse space problem) $G^T D^{-1} G$, which can be expensive. Further, factoring and solving the coarse problem can represent a serious impediment to parallel scalability [3].

Instead, recent research in KKT preconditioners explores the use of an inexact Schur complement matrix, which can typically be computed at greatly reduced cost [20, 33, 43]. Applying results from sections 4.5 and 4.6, we modify the block diagonal preconditioner (4.3.2) to use an inexact Schur complement matrix

$$\begin{bmatrix} D^{-1} & 0 \\ 0 & \left(G^T D^{-1} G\right)^{-1} \end{bmatrix} \approx \begin{bmatrix} L_D^T L_D & 0 \\ 0 & L_S^T L_S \end{bmatrix}, \tag{4.6.1}$$

where

$$\begin{aligned} D^{-1} &= L_D^T L_D \\ \left(G^T D^{-1} G\right)^{-1} &\approx L_S^T L_S, \end{aligned}$$

and

$$L_S \left(G^T D^{-1} G\right) L_S^T = I + \mathcal{E}.$$

If we precondition the dual interface problem (4.2.5) with the preconditioner (4.6.1), we

arrive at the preconditioned system [43]

$$\begin{bmatrix} \widetilde{F} & -\widehat{G} \\ -\widehat{G}^T & 0 \end{bmatrix} \begin{bmatrix} \widetilde{\lambda} \\ \widehat{\alpha} \end{bmatrix} = \begin{bmatrix} \widetilde{d} \\ -\widehat{e} \end{bmatrix}. \tag{4.6.2}$$

We cannot simply apply PCPG at this point, as it would require the inverse of the Schur complement, which we are trying to avoid [7, 43]. We instead split the linear system using a different splitting [43]. This produces

$$\begin{bmatrix} I & -\widehat{G} \\ -\widehat{G}^T & \mathcal{E} \end{bmatrix} \begin{bmatrix} \widetilde{\lambda} \\ \widehat{\alpha} \end{bmatrix} = \begin{bmatrix} \widetilde{S} & 0 \\ 0 & \mathcal{E} \end{bmatrix} \begin{bmatrix} \widetilde{\lambda} \\ \widehat{\alpha} \end{bmatrix} \begin{bmatrix} \widetilde{d} \\ -\widehat{e} \end{bmatrix},$$

where $\widetilde{F} = I - \widetilde{S}$, and $\widehat{G}^T\widehat{G} = I + \mathcal{E}$. Multiplying through by the inverse of the matrix on the left-hand side gives the fixed-point iteration

$$\begin{bmatrix} \widetilde{\lambda}^{(k+1)} \\ \widehat{\alpha}^{(k+1)} \end{bmatrix} = \begin{bmatrix} \widehat{P}\widetilde{S} & -\widehat{G}\mathcal{E} \\ -\widehat{G}^T\widetilde{S} & -\mathcal{E} \end{bmatrix} \begin{bmatrix} \widetilde{\lambda}^{(k)} \\ \widehat{\alpha}^{(k)} \end{bmatrix} + \begin{bmatrix} \widehat{P}\widetilde{d} + \widehat{G}\widehat{e} \\ -\widehat{G}^T\widetilde{d} + \widehat{e} \end{bmatrix},$$

where in this case $\widehat{P} = I - \widehat{G}\widehat{G}^T$ is *not* a projector. Writing the linear system for the fixed-point gives the related system

$$\begin{bmatrix} I - \widehat{P}\widetilde{S} & \widehat{G}\mathcal{E} \\ \widehat{G}^T\widetilde{S} & I + \mathcal{E} \end{bmatrix} \begin{bmatrix} \widetilde{\lambda} \\ \widehat{\alpha} \end{bmatrix} = \begin{bmatrix} \widehat{P}\widetilde{d} + \widehat{G}\widehat{e} \\ -\widehat{G}^T\widetilde{d} + \widehat{e} \end{bmatrix}. \tag{4.6.3}$$

Unlike the case with an exact Schur complement, we cannot reduce the size of the system to be solved. However, especially for 3D models, the size increase is very modest.

By combining [43, Theorem 4.2] with the observation that $\|\widehat{G}\|_2^2 = \|I + \mathcal{E}\|_2^2$, the eigen-

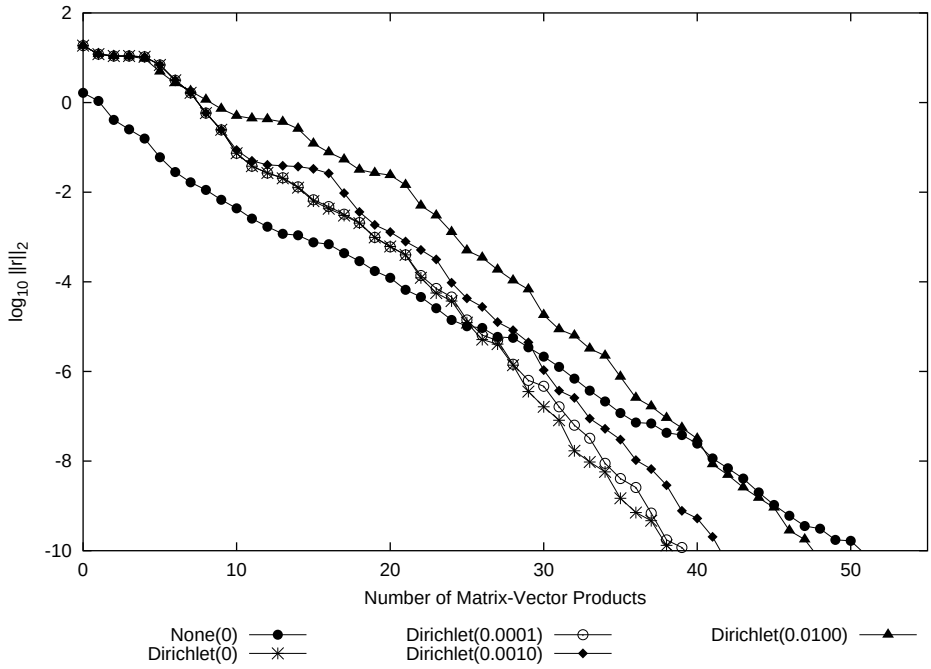


Figure 4.6.1: Number of matrix-vector products vs. residual norm using an approximate Schur complement formulation of FETI. The number in parentheses indicates the drop tolerance used, with (0) indicating the exact Schur complement.

values μ of the matrix in (4.6.3) can be bounded about one:

$$|1-\mu| \leq \left(1+\|I+\mathcal{E}\|_2^2\right) \cdot \max\left(\|\widetilde{S}\|_2, \|\mathcal{E}\|_2\right). \tag{4.6.4}$$

Here, $\|\widetilde{S}\|_2$ is a measure of the accuracy of the FETI preconditioner, and $\|\mathcal{E}\|_2$ is a measure of the accuracy of the approximate Schur complement. This bound suggests that using a less expensive approximation to the inverse of the Schur complement may not significantly impact the eigenvalue distribution, and thus the overall convergence rate of an iterative method. In particular, the theory presented here can be used to estimate the impact on the eigenvalue distribution and the convergence rate of the iterative method in the case where we "recycle" a factored Schur complement from a previous linear system in a sequence.

We present a simple example illustrating the impact of an approximate Schur complement on convergence. A 2D finite element model of a cantilevered beam was cut into a

3×3 decomposition of 9 subdomains, and the resulting dual problem solved by applying full GMRES to the related system (4.6.3). An approximate Schur complement was generated using an incomplete Cholesky decomposition with a drop tolerance [42]. While not a practical choice, it gives the ability to vary the accuracy of the approximation. In Figure 4.6.1, we show convergence curves for an exact Schur complement with a Dirichlet preconditioner and with no preconditioner, and also an approximate Schur complement with a Dirichlet preconditioner. In the figure, the number in parentheses indicates the drop tolerance used, with (0) indicating the exact Schur complement. This demonstrates the interplay between the Schur complement and preconditioner, as suggested in the bound (4.6.4). In the presence of a strong preconditioner, it is possible to use an inexact Schur complement without significantly impacting convergence.

4.7 Conclusions

We have shown that every FETI preconditioner may be viewed as a splitting of the $(1,1)$ block of (4.2.5), suggesting algebraic (rather than mechanical) means of constructing and analyzing new FETI preconditioners. How to choose a better splitting $F = D - E$ (e.g., a better preconditioner) requires further investigation. We have also shown equivalence between a class of block-diagonal preconditioners and traditional FETI preconditioners. These new algebraic connections make existing KKT preconditioner analysis immediately applicable to FETI systems. We have leveraged this analysis to provide a theory regarding the clustering of the eigenvalues of preconditioned systems, which provides a mechanism to evaluate existing and new FETI preconditioners. Furthermore, the use of a preconditioner based on an approximate Schur complement may be computationally more efficient than existing FETI preconditioners, especially when solving a sequence of FETI systems where the Schur complement matrix can be recycled between systems.

Chapter 5

Conclusions

We have discussed improvements to solvers and preconditioners for sequences of linear systems arising in nonlinear finite element analysis, with a focus on recycling information between consecutive linear systems in a sequence. In Chapter 2 we discussed the theory of Krylov subspace recycling, and introduced two linear solvers, GCRO-DR and a modification of GCROT to support recycling. When solving a sequence of linear systems, methods employing Krylov subspace recycling frequently outperformed GMRES while keeping only a small number of vectors, although this was not always true.

Chapter 3 presented a bound on the convergence of GCRO-DR using deflation-based Krylov subspace recycling. The analysis shows that if the recycled subspace $\mathcal{R}(C_k)$ contains an invariant subspace $\mathcal{R}(Q_\ell)$, convergence can be bounded above by problem (3.3.3). We performed numerical experiments to evaluate the usefulness of these bounds, and found them to be tight in cases where $\|P_Q\|_2$ is not large. Experimental evidence shows that a deflationary approach is not optimal, and that there exist better choices than simply trying to deflate the eigenvalues closest to the origin. More work is needed to determine how to identify and select better subspaces within the recycling process.

In Chapter 4 we turned to analysis of preconditioners for FETI systems, and showed that every FETI preconditioner may be viewed as a splitting of the $(1,1)$ block of the FETI dual-interface problem, suggesting algebraic (rather than mechanical) means of construct-

ing and analyzing new FETI preconditioners. We have also shown the equivalence between a class of block-diagonal preconditioners and traditional FETI preconditioners. Further, we showed equivalence between the related system and the FETI projected interface problem. We supply bounds on the eigenvalues of preconditioned FETI systems, which provides a mechanism to evaluate existing and new FETI preconditioners. Finally, we demonstrated that the use of a preconditioner based on an approximate Schur complement may not significantly impact convergence, and has the potential to be computationally less expensive. This may be especially beneficial when solving a sequence of FETI systems where the Schur complement matrix can be recycled between systems.

Appendix A

Algorithm A.1: **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(1):243–269, January 1999.

[2] Christopher Beattie, Mark Embree, and John Rossi. Convergence of restarted Krylov subspaces to invariant subspaces. *SIAM Journal on Matrix Analysis and Applications*, 25:1074–1109, 2004.

[3] Manoj Bhardwaj, David Day, Charbel Farhat, Michel Lesoinne, Kendall Pierson, and Daniel Rixen. Application of the FETI method to ASCI problems - scalability results on one thousand processors and discussion of highly heterogeneous problems. *Int. J. Numer. Meth. Engrg.*, 47:513–535, 2000.

[4] Ronald F. Boisvert, Roldan Pozo, Karin Remington, Richard F. Barrett, and Jack J. Dongarra. Matrix Market: A Web resource for test matrix collections. In Ronald F. Boisvert, editor, *Quality of Numerical Software: Assessment and Enhancement*, pages 125–136. Chapman and Hall , London, 1997.

[5] Tony F. Chan and Michael K. Ng. Galerkin projection methods for solving multiple linear systems. *SIAM Journal on Scientific Computing*, 21(3):836–850, May 1999.

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

[7] E. de Sturler and Jorg Liesen. Block-diagonal preconditioners for indefinite linear algebraic systems, part I: Theory. Technical Report UIUCDCS-R-2002-2279, University of Illinois at Urbana-Champaign, Urbana, IL, 2002.

[8] Eric de Sturler. Nested Krylov methods based on GCR. *Journal of Computational and Applied Mathematics*, 67:15–41, 1996.

[9] Eric de Sturler. Truncation strategies for optimal Krylov subspace methods. *SIAM Journal on Numerical Analysis*, 36(3):864–889, June 1999.

[10] Jack J. Dongarra, Iain S. Duff, Danny C. Sorensen, and Henk A. van der Vorst. *Numerical linear algebra for high-performance computers*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 1998.

[11] Michael Eiermann, Oliver G. Ernst, and Olaf Schneider. Analysis of acceleration strategies for restarted minimal residual methods. *Journal of Computational and Applied Mathematics*, 123:261–292, 2000.

[12] Stanley C. Eisenstat, Howard C. Elman, and Martin H. Schultz. Variational iterative methods for nonsymmetric systems of linear equations. *SIAM Journal on Numerical Analysis*, 20(2):345–357, April 1983.

[13] Howard C. Elman, David. J. Silvester, and Andrew J. Wathen. Iterative methods for problems in computational fluid dynamics. In R. Chan, T. Chan, and G. Golub, editors, *Iterative Methods in Scientific Computing*, pages 271–327. Springer-Verlag, Singapore, 1997.

[14] J. Erhel, K. Burrage, and B. Pohl. Restarted GMRES preconditioned by deflation. *Journal of Computational and Applied Mathematics*, 69(5):303–318, 1996.

[15] C. Farhat, K. H. Pierson, and M. Lesoinne. The second generation of FETI methods and their application to the parallel solution of large-scale linear and geometrically nonlinear structural analysis problems. *Computer Methods in Applied Mechanics and Engineering*, 184:333–374, 2000.

[16] Charbel Farhat and François-Xavier Roux. Implicit parallel processing in structural mechanics. In J. Tinsley Oden, editor, *Computational Mechanics Advances*, volume 2 (1), pages 1–124. North-Holland, 1994.

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

[18] Gene H. Golub and Charles F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, Baltimore, MD, USA, third edition, 1996.

[19] 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*, 79:553–575, 2001.

[20] J. C. Haws and C. D. Meyer. Preconditioning KKT systems. Numer. Linear Algebra Appl., to appear, October 2002.

[21] D.D. Johnson, D.M. Nicholson, F.J. Pinski, B.L. Gyorffy, and G.M. Stocks. Energy and pressure calculations for random substitutional alloys. *Physical Review B*, 41(14):9701–9716, May 1990.

[22] W. Kohn and N. Rostoker. Solution of the Schrödinger equation in periodic lattices with an application to metallic lithium. *Physical Review*, 94(5):1111–1120, June 1954.

[23] J Korringa. On the calculation of the energy of a Bloch wave in a metal. *Physica*, XIII(1):392–400, 1947.

[24] F. J. Lingen and M. G. A. Tijssens. An efficient parallel procedure for the simulation of crack growth using the cohesive surface methodology. *Int. J. Numer. Meth. Engng.*, 52:867–888, 2001.

[25] Greg Mackey. Reusing Krylov subspaces for sequences of linear systems. Master's thesis, University of Illinois at Urbana-Champaign, 2003.

[26] Jan Mandel and Radek Tezaur. Convergence of a substructuring method with Lagrange multipliers. *Numerische Mathematik*, 73(4):473–487, June 1996.

[27] Björn Medeke. Set QCD: Quantum Chromodynamics. Description of matrix set on NIST Matrix Market. http://math.nist.gov/MatrixMarket.

[28] G. A. Meurant. *Computer Solution of Large Linear Systems*, volume 28 of *Studies in Mathematics and Its Applications*. North-Holland, Amsterdam, 1999.

[29] Ronald B. Morgan. A restarted GMRES method augmented with eigenvectors. *SIAM Journal on Matrix Analysis and Applications*, 16(4):1154–1171, October 1995.

[30] Ronald B. Morgan. Implicitly restarted GMRES and Arnoldi methods for nonsymmetric systems of equations. *SIAM Journal on Matrix Analysis and Applications*, 21(4):1112–1135, October 2000.

[31] Ronald B. Morgan. GMRES with deflated restarting. *SIAM Journal on Scientific Computing*, 24(1):20–37, January 2003.

[32] Dianne P. O'Leary. The block conjugate gradient algorithm and related methods. *Linear Algebra and its Applications*, 29:293–322, 1980.

[33] I. Perugia and V. Simoncini. Block-diagonal and indefinite symmetric preconditioners for mixed finite element formulations. *Numerical linear algebra with applications*, 7(7–8):585–616, October/December 2000.

[34] Christian Rey and Franck Risler. A Rayleigh-Ritz preconditioner for the iterative solution to large scale nonlinear problems. *Numerical Algorithms*, 17(3–4):279–311, 1998.

[35] Franck Risler and Christian Rey. On the reuse of Ritz vectors for the solution to nonlinear elasticity problems by domain decomposition methods. *Contemporary Mathematics*, 218:334–340, 1998.

[36] Franck Risler and Christian Rey. Iterative accelerating algorithms with Krylov subspaces for the solution to large-scale nonlinear problems. *Numerical Algorithms*, 23(1):1–30, 2000.

[37] Daniel Rixen and Charbel Farhat. A simple and efficient extension of a class of substructure based preconditioners to heterogeneous structural mechanics problems. *Inter. J. Numer. Meth. Engrg.*, 44:489–516, 1999.

[38] F. X. Roux. Spectral analysis of the interface operators associated with the preconditioned saddle-point principle domain decomposition method. In David E. Keyes, Tony F. Chan, Gérard A. Meurant, Jeffrey S. Scroggs, and Robert G. Voigt, editors, *Proceedings of the Fifth International Symposium on Domain Decomposition Methods for Partial Differential Equations*, pages 73–89, Philadelphia, PA, 1992. SIAM. Held in Norfolk, VA, May 6–8, 1991.

[39] Y. Saad, M. Yeung, J. Erhel, and F. Guyomarc'h. A deflated version of the Conjugate Gradient algorithm. *SIAM Journal on Scientific Computing*, 21(5):1909–1926, September 2000.

[40] Youcef Saad and Martin H. Schultz. GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal on Scientific and Statistical Computing*, 7(3):856–869, July 1986.

[41] Yousef Saad. Analysis of augmented Krylov subspace methods. *SIAM Journal on Matrix Analysis and Applications*, 18(2):435–449, April 1997.

[42] Yousef Saad. *Iterative methods for sparse linear systems*. Society for Industrial and Applied Mathematics, Philadelphia, PA, second edition, 2003.

[43] Chris Siefert and Eric de Sturler. Preconditioners for generalized saddle-point problems. Technical Report UIUCDCS-R-2004-2448, University of Illinois at Urbana-Champaign, June 2004. Submitted to SIAM Journal on Numerical Analysis.

[44] V. Simoncini and E. Gallopoulos. An iterative method for nonsymmetric systems with multiple right-hand sides. *SIAM J. Sci. Comput.*, 16:917–933, 1995.

[45] Valeria Simoncini and Daniel Szyld. On the superlinear convergence of exact and inexact Krylov subspace methods. Technical Report 03-3-13, Temple University, March 2003. Revised December 2003.

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

[47] Barry F. Smith, Petter E. Bjørstad, and William D. Gropp. *Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations*. Cambridge Univ. Press, may 1996.

[48] G. W. Stewart and Ji-guang Sun. *Matrix Perturbation Theory*. Academic Press, San Diego, 1990.

[49] Zdeněk Strakoš and Petr Tichý. On error estimation in the conjugate gradient method and why it works in finite precision computations. *ETNA*, 13:56–80, 2002.

[50] Radek Tezaur. *Analysis of Lagrange Multiplier Based Domain Decomposition*. PhD thesis, University of Colorado at Denver, Department of Mathematics, 1998.

[51] 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. Commun.*, 146:203–224, 2002.

[52] 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.

[53] U. Meier Yang and K. A. Gallivan. A new family of block methods. *Applied Numerical Mathematics: Transactions of IMACS*, 30(2–3):155–173, June 1999.

[54] R Zeller, P.H. Dederichs, B. Ujfalussy, L. Szunyogh, and P. Weinberger. Theory and convergence properties of the screened Korringa-Kohn-Rostoker method. *Physical Review B*, 52(12):8807–8812, September 1995.

Author’s Biography

Michael Parks was born in Knoxville, Tennessee, on November 25, 1975. He obtained bachelor’s degrees in computer science and physics from the Virginia Polytechnic Institute and State University (Virginia Tech) in 1998, and a masters degree in computer science in 2000. He began his graduate studies at the University of Illinois at Urbana-Champaign in 2000, and received his Ph.D. in 2005. In the fall of 2004, he joined the staff at Sandia National Laboratories in Albuquerque, New Mexico.