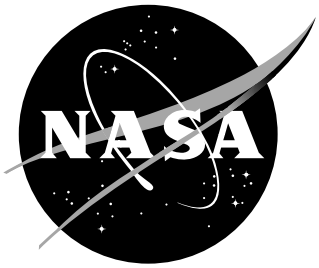
NASA/TM–2010–216190



A General Algorithm for Reusing Krylov Subspace Information. I. Unsteady Navier-Stokes

Mark H. Carpenter
Langley Research Center, Hampton, Virginia

C. Vuik, Peter Lucas, Martin van Gijzen, and Hester Bijl
Delft University of Technology, Delft, The Netherlands

January 2010

NASA STI Program ... in Profile

Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA scientific and technical information (STI) program plays a key part in helping NASA maintain this important role.

The NASA STI Program operates under the auspices of the Agency Chief Information Officer. It collects, organizes, provides for archiving, and disseminates NASA's STI. The NASA STI Program provides access to the NASA Aeronautics and Space Database and its public interface, the NASA Technical Report Server, thus providing one of the largest collection of aeronautical and space science STI in the world. Results are published in both non-NASA channels and by NASA in the NASA STI Report Series, which includes the following report types:

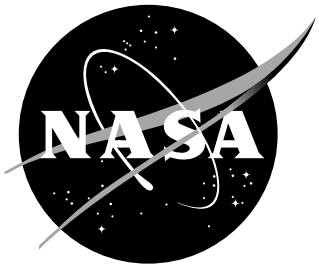
- TECHNICAL PUBLICATION. Reports of completed research or a major significant phase of research that present the results of NASA programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA counterpart of peer-reviewed formal professional papers, but having less stringent limitations on manuscript length and extent of graphic presentations.
- TECHNICAL MEMORANDUM. Scientific and technical findings that are preliminary or of specialized interest, e.g., quick release reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.
- CONTRACTOR REPORT. Scientific and technical findings by NASA-sponsored contractors and grantees.
- CONFERENCE PUBLICATION. Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or co-sponsored by NASA.
- SPECIAL PUBLICATION. Scientific, technical, or historical information from NASA programs, projects, and missions, often concerned with subjects having substantial public interest.
- TECHNICAL TRANSLATION. English-language translations of foreign scientific and technical material pertinent to NASA's mission.

Specialized services also include creating custom thesauri, building customized databases, and organizing and publishing research results.

For more information about the NASA STI Program, see the following:

- Access the NASA STI program home page at ***http://www.sti.nasa.gov***
- E-mail your question via the Internet to help@sti.nasa.gov
- Fax your question to the NASA STI Help Desk at 443-757-5803
- Phone the NASA STI Help Desk at 443-757-5802
- Write to:
  NASA STI Help Desk
  NASA Center for AeroSpace Information
  7115 Standard Drive
  Hanover, MD 21076–1320

NASA/TM–2010–216190



A General Algorithm for Reusing Krylov Subspace Information. I. Unsteady Navier-Stokes

Mark H. Carpenter
Langley Research Center, Hampton, Virginia

C. Vuik, Peter Lucas, Martin van Gijzen, and Hester Bijl
Delft University of Technology, Delft, The Netherlands

National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23681-2199

January 2010

Acknowledgments

Performed while in residence at Technical University of Delft, The Netherlands. Technical monitor Prof. Hester Bijl. Special thanks are extended to the faculty and staff of TUDelft, for many stimulating discussions on this and other topics.

Available from:

NASA Center for AeroSpace Information
7115 Standard Drive
Hanover, MD 21076-1320
443-757-5802

Abstract

A general algorithm is developed that reuses available information to accelerate the iterative convergence of linear systems with multiple right-hand sides $\mathcal{A}\ \mathbf{x} = \mathbf{b}^i$, which are commonly encountered in steady or unsteady simulations of nonlinear equations. The algorithm is based on the classical GMRES algorithm with eigenvector enrichment but also includes a Galerkin projection preprocessing step and several novel Krylov subspace reuse strategies. The new approach is applied to a set of test problems, including an unsteady turbulent airfoil, and is shown in some cases to provide significant improvement in computational efficiency relative to baseline approaches.

Contents

1 Introduction

Krylov-based methods (e.g., GCR [1], GMRES [2], BI-CGSTAB [3], FGMRES [4], and GMRESR [5]) have emerged as popular methods for solving large, indefinite systems $\mathcal{A}\,\mathbf{x} = \mathbf{b}$. We can argue that their use is now ubiquitous throughout the majority of the mathematical, scientific, and engineering disciplines. Indeed, the fluid mechanics community has a long history of using Krylov methods. See, for example, references [6–13] as well as the references therein.

Krylov deflation and enrichment techniques are generalizations of the aforementioned basic techniques, that in some instances greatly extend the efficiency and robustness of conventional approaches. These techniques glean "important" information (e.g., problematic eigenvectors) from past problems to enrich the current subspace. As a result, they are well suited for slowly varying, recursive problems of the form $\mathcal{A}^{\,i}\,\mathbf{x} = \mathbf{b}^{\,i}$.

That said, Krylov methods (both basic and generalized) have not significantly penetrated the software used to perform large-scale aerodynamic engineering simulations. Fixed-point iterations (e.g., basic iterative methods or multigrid methods that use basic iterative methods as smoothers) are still used extensively for these calculations. The lack of penetration of Krylov methods into the production aerodynamic arena is not without reason. In addition to the inevitable inertia that is related to modification of legacy software, other impediments include

- Memory management: Large, complex simulations (e.g., adjoint, moving grid, and adaptive grid) already require significant memory, which leaves little room to store additional Krylov vectors.
- Algorithmic complexity: The implementation of succinctly written pseudo-code in legacy software which involves millions of lines, is far from trivial.
- Computer science: Legacy fixed-point solvers are highly optimized (e.g., data structures and cache); thus comparison of existing methods with new methods can be difficult.

Our goals herein are twofold. First, we assemble a general purpose "GMRES type" algorithm that is tailored for problems of the form $\mathcal{A}^{\,i}\,\mathbf{x} = \mathbf{b}^{\,i}$; the algorithm is simple, efficient, and robust but flexible enough to accommodate the nuances that commonly arise in large-scale computations (e.g., nonlinear steady and unsteady simulations). To meet these challenges, we propose an algorithm that features a Krylov subspace enrichment technique that can accommodate *arbitrary*

enrichment vectors to provide extreme flexibility. The algorithm is enhanced with a subelement that permits the incorporation of solutions from past problems, a technique that is shown to complement conventional eigenvector subspace enrichment. Finally, several implementation details are addressed that enhance the usability of the algorithm (e.g., a no-cost residual estimator and a low-cost eigenvector error estimator). Although many of the individual elements of the algorithm are not new, the kernel algorithm and the supporting subelements result in a simple yet general algorithm that we feel is a useful improvement over other deflation and enrichment techniques in the literature.

Our second goal is to apply the general-purpose algorithm (using legacy software) to large-scale fluid mechanics applications and to establish its effectiveness in routine settings that are encountered by practitioners (e.g., ill-conditioned systems that involve $> 10^7$ unknowns). Satisfying this goal necessarily requires testing various equation sets, problem sizes, preconditioners and termination tolerances, all in the context of different enrichment strategies.

After we complete the preliminary testing on model three-dimensional (3D) advection-diffusion problems, we test the newly proposed algorithm on a classical Newton-Krylov nonlinear iteration. The testing is performed in the context of the Hexstream [14, 15] software, using the 2D, unsteady RANS (URANS) equations to solve for flow past a wind-turbine blade at a high angle of attack. The principle focus is to establish the efficacy of enrichment on this class of unsteady problems.

The paper is organized as follows. Section II presents a literature review of deflation and enrichment techniques. Section III presents for the purpose of reference, the conventional GMRES algorithm. Section IV presents the generalized enrichment algorithm and related subelements. Included are proofs of the optimality of the new algorithm, as well as its initialization step. Section V presents the set of large-scale linear and nonlinear test problems, while section VI presents the results for all test cases, followed by conclusions. The first Appendix provides a proof that the proposed enrichment technique builds a Krylov subspace for the portion of the space that is appended to the enrichment vectors. A second Appendix provides the details for approximating eigenvectors by using conventional and harmonic Ritz approximations.

2 The Case for Krylov Subspace Deflation/Enrichment

A simple and frequently effective remedy for slow convergence or stall is to remove: *deflate* the troublesome eigenvalues (eigenvectors) from the problem. Deflation relies on the construction of an auxiliary approximation subspace that is designed to target the problematic modes. Numerous techniques can be used to deflate problematic eigenvectors. They differ primarily in the method of implementation, (i.e., how the problematic eigenvectors are introduced into the base algorithm) and the manner in which the eigenvectors or problematic modes are approximated and constructed.

Deflation techniques essentially can be implemented in one of two ways. In the first approach, called *deflation*, the underlying matrix is modified to eliminate the problematic eigenvalues. For example, a projection preconditioner $\mathcal{P} = \mathcal{I} - \mathcal{A}\,\mathcal{Z}_k \left(\mathcal{Z}_k^T\,\mathcal{A}\,\mathcal{Z}_k\right)^{-1} \mathcal{Z}_k^T$ is constructed from the offending eigenvectors $\mathcal{Z}_k$ and is used to modify the system into the form

$$\mathcal{P}\ [\ \mathcal{A}\ \mathbf{x}\ -\ \mathbf{b}\ ]\ =\ 0\quad .$$

A conventional minimization algorithm (e.g., GMRES) is then used to solve the preconditioned system.

In the second approach, called *augmentation* or *enrichment*, the offending eigenvectors $\mathcal{Z}_k$ are added directly to the search space, (e.g., $span\{\mathcal{Z}_k, \mathcal{Z}_{m-k}\}$). The first k vectors contain important information about the problematic eigenmodes, while the remaining vectors $z_1, ..., z_{m-k}$ are determined by using the standard Arnoldi procedure started from the current residual.

The literature contains a wealth of techniques that advocate some form of deflation or enrichment. We begin with those deflation techniques that are similar to those originally presented in the work of Shroff and Keller [16–22]. Shroff and Keller [16] developed a technique to stabilize unstable procedures by dividing the iteration into an orthogonal projection step that is handled by Newton iteration; its complement is handled with a fixed-point iteration. Similarly, Kharchenko and Yeremin [17] form a preconditioner that modifies the matrix $\mathcal{A}$ with approximate eigenvectors that are used to translate a group of small eigenvalues by means of a series of low-rank projections. Erhel, Burrage, and Pohl [18] used approximate eigenvectors to build a preconditioner that shifts eigenvalues. This method was further modified by Burrage et al. [20] and Burrage and Erhel [21] to include a harmonic Ritz procedure and other methods for isolating the eigenvalues. Baglama, Calvetti, Golub, and Reichel [22] extend earlier works of Erhel et al. [18] by using an implicit restarting approach that is similar to the implicitly restarted Arnoldi (IRA) method of Sorensen [23].

Augmentation, or enrichment techniques were developed by Morgan [24–26] and Morgan and Zeng [27]. Morgan developed a method for augmenting the Krylov space with eigenvectors (GMRES-EN) [24]. This approach is a powerful method for problems with small eigenvalues and works best when the matrix vector costs are high. Like other deflation techniques, this method does not work as well when small eigenvalues are less separated from zero than from each other. Morgan improved the GMRES-EN method [25] by incorporating ideas from Sorensen's [23] IRA algorithm, thereby allowing the implicit restarting of the GMRES algorithm. Morgan developed implicitly restarted GMRES-DR algorithm [26] by using existing harmonic Ritz vectors that are prepended into the space. In so doing, however, the algorithm becomes valid only for harmonic Ritz vectors.

Over the past decade, all of these algorithms have converged in their evolution with the incorporation of similar ideas. Chapman and Saad [19] presented a general framework for deflation and augmentation approaches. Their enrichment approach consists of appending the problematic eigenvectors at the end of the Krylov space. Saad [28] derives residual norm estimates for a general class of methods that are based on projection techniques for subspaces of the form $K_m + W$, where K_m is the standard Krylov subspace and W is an arbitrary subspace.

Several deflation/enrichment approaches have been developed that are tailored specifically for solving problems of the form $\mathcal{A}^{\,i}\,\mathbf{x} \;=\; \mathbf{b}^{\,i}$. (The superscript i denotes distinct right-hand side (RHS) vectors.) A large number of block methods are available to deal with multiple, coexisting RHS vectors [29–34]. Our problems of interest do not have simultaneously available RHS data. The literature on consecutive RHS acceleration is lacking. Erhel and Guyomarc'h [35] devise a modified conjugate gradient (CG) method that uses orthogonal projections to recycle Krylov subspace information. Golub et al. [36] combine CG with Chebyshev filtering polynomials to build a preconditioner that is well suited for multiple RHS. Parks et al. [37] propose a new solver, GCRO-DR, to implement Krylov subspace recycling of approximate invariant subspaces and to provide theoretical results to analyze the convergence of GCRO-DR for a typical application that generates a long sequence of linear systems.

The literature for nonlinear applications (e.g., $\mathcal{A}^{\,i}\,\mathbf{x} \;=\; \mathbf{b}^{\,i}$ as it arises from a Newton-type

iteration) is also sparse. Rey and Risler [38] develop a Rayleigh-Ritz preconditioner for use with CG in the context of nonlinear problems. The nonlinear problem is solved with a Newton-Raphson iteration and requires the solution of a sequence of related linear problems $\mathcal{A}^{\,i}\,\mathbf{x} \;=\; \mathbf{b}^{\,i}$. This projection preconditioner is constructed and then used on subsequent problems. The approach is further refined by Risler and Rey in reference [39]; several new IRKS (iterative reuse of Krylov subspaces) algorithms are presented for the CG method. Their focus is on developing super-subspace information based on Krylov information that is gleaned from past problems, for use in accelerating the current iteration. Tromeur-Dervout and Vassilevsi [40] formulate a nonlinear, reduced model technique for computing a better initial guess for the inexact Newton method.

Finally, we note that many problems of practical interest result in nonsymmetric matrices of large dimensionality. To date, symmetric problems have received much more attention in the deflation and augmentation literature than have nonsymmetric problems. Furthermore, the test problems that are used in the deflation literature are typically of low dimensionality. Therefore, the effects of asymmetry and large dimensionality, clearly must be included in both the present and future deflation studies.

3 Krylov Subspace Methods: Background

3.1 Nomenclature

The derivation and analysis of Krylov subspace methods relies heavily on vector theory, and the theory of square and non-square matrices. Herein, matrices are presented in italics letters (e.g., $\mathcal{A}$, $\mathcal{H}_m$, $\bar{\mathcal{H}}_m$), with the "over-bar" nomenclature: (e.g., $\bar{\mathcal{H}}_m$) reserved for non square matrices. The subscript signifies the column dimension, while the row dimension is implied from the context of the equation.

Vectors are presented in bold letters (e.g., $\mathbf{b}$, $\mathbf{r}_m$, $\mathbf{x}$). Groups of vectors (vector spaces) are presented in standard text, with a subscript signifying the dimensionality of the space (e.g., V_m , S_m , V_{m+1}). The vector spaces routinely manipulated as matrices.

With a slight abuse of nomenclature, iteration numbers are also denoted using subscripts (e.g., $\mathbf{x}_m$, $\mathbf{r}_m$). Further clarification is given if ambiguity exists in the meaning of the subscript. Matrices and vectors that do not have subscripts are of dimension N.

3.2 Conventional GMRES

A general version of GMRES with enrichment is developed, and is shown to be efficient for solving large systems of slowly varying linear equations. The algorithm is an extension of the well known GMRES [2] algorithm. It is therefore instructive to first describe the important elements of GMRES.

Let $\mathcal{A}$ be an $N \times N$ matrix, and $\mathbf{b}$ a vector of length N, that define the linear system

$$\mathcal{A}\;\mathbf{x} \;=\; \mathbf{b} \quad . \tag{1}$$

Choose an arbitrary starting guess $\mathbf{x}_0$ and form the initial residual $\mathbf{r}_0 \;=\; \mathbf{b} \;-\; \mathcal{A}\;\mathbf{x}_0$. Next, generate using the Arnoldi process, the orthogonal basis V_m that spans the m dimensional Krylov space

$$\mathcal{K}_m \;=\; span\{\,\mathbf{r}_0\,,\,\mathcal{A}\;\mathbf{r}_0\,,\,\mathcal{A}^{\,2}\,\mathbf{r}_0\,,\cdots,\,\mathcal{A}^{\,m-1}\,\mathbf{r}_0\,\} \quad . \tag{2}$$

Consequently, V_m satisfies the relationship

$$\mathcal{A}\ V_m \quad = \quad V_m\ \mathcal{H}_m\ +\ \mathbf{v}_{m+1}\ h_{m+1,m}\,{\mathbf{e}_m}^T \quad = \quad V_{m+1}\ \bar{\mathcal{H}}_m \tag{3}$$

$$[\,V_{m+1}\,]^T\ V_{m+1}\ =\ \mathcal{I}_{m+1} \tag{4}$$

with the unit coordinate vector $\mathbf{e}_m \in \mathcal{R}^m$, defined by $\mathbf{e}_m = [0, \cdots, 0, 1]^T$, $\bar{\mathcal{H}}_m$ an upper Hessenberg matrix, and $\mathcal{I}_{m+1}$ the identity matrix of dimension m.

The GMRES algorithm builds an approximate solution to equation (1) of the form

$$\mathbf{x}_m\ =\ \mathbf{x}_0\ +\ V_m\ \mathbf{y}_m \tag{5}$$

with the vector $\mathbf{y}_m\ \in \mathcal{R}^m$, and determined such that the L_2 norm of the residual $\mathbf{r}_m\ =\ \mathbf{b} - \mathcal{A}\,\mathbf{x}_m$ is minimized over $\mathcal{K}_m$. To illustrate, define $\beta\ =\ \|\,\mathbf{r}_0\,\|_2$, and $\mathbf{v}_1\ =\ \mathbf{r}_0\,/\beta$, then substitute equation (5) into $\mathbf{r}_m\ =\ \mathbf{b}\ -\ \mathcal{A}\ \mathbf{x}_m$. Simplifying the resulting expression yields

$$\begin{aligned}\mathbf{r}_m\ =\ \mathbf{b}\ -\ \mathcal{A}\ \mathbf{x}_m\ =\ \mathbf{r}_0\ -\ \mathcal{A}\ V_m\ \mathbf{y}_m \quad &= \quad \mathbf{r}_0\ -\ V_{m+1}\ \bar{\mathcal{H}}_m\ \mathbf{y}_m \\ &= \quad \mathbf{v}_1\,\beta - V_{m+1}\ \bar{\mathcal{H}}_m\ \mathbf{y}_m\ =\ V_{m+1}\,(\beta\,\mathbf{e}_1\ -\ \bar{\mathcal{H}}_m\ \mathbf{y}_m\,)\end{aligned} \tag{6}$$

By construction, V_{m+1} is orthogonal, and the L_2 norm of $\mathbf{r}_m$ simplifies to $\|\,\mathbf{r}_m\,\|_2 = \|\beta\,\mathbf{e}_1\ -\ \bar{\mathcal{H}}_m\ \mathbf{y}_m\,\|_2$. The GMRES algorithm chooses the vector $\mathbf{y}_m$ such that the residual $\|\,\mathbf{r}_m\,\|_2$ is minimized; i.e., $\mathbf{y}_m\ =\ \mathrm{argmin}_y\|\beta\,\mathbf{e}_1\ -\ \bar{\mathcal{H}}_m\ \mathbf{y}_m\,\|_2$.

To solve for $\mathbf{y}_m$, first form the $Q\,R$ decomposition of the matrix $\bar{\mathcal{H}}_m\ =\ \mathcal{Q}_{m+1}\ \bar{\mathcal{R}}_m$, with ${\mathcal{Q}_{m+1}}^T\,\mathcal{Q}_{m+1}\ =\ \mathcal{Q}_{m+1}\ {\mathcal{Q}_{m+1}}^T\ =\ \mathcal{I}_{m+1}$. Alternatively, note that $\bar{\mathcal{H}}_m\ =\ \bar{\mathcal{Q}}_m\ \mathcal{R}_m$ with ${\bar{\mathcal{Q}}_m}^T\,\bar{\mathcal{Q}}_m\ =\ \mathcal{I}_m$. (The matrix $\bar{\mathcal{R}}_m$ is obtained from $\mathcal{R}_m$ by appending the zero transpose vector in row $m+1$.) Next, substitute $\bar{\mathcal{H}}_m$ into the residual $\|\,\mathbf{r}_m\,\|_2$ to obtain

$$\|\,\mathbf{r}_m\,\|_2\ =\ \|\beta\,\mathcal{Q}_{m+1}\,(\,{\mathcal{Q}_{m+1}}^T\,\mathbf{e}_1\ -\ \bar{\mathcal{R}}_m\ \mathbf{y}_m\,)\|_2\ =\ \|\beta\,{\mathcal{Q}_{m+1}}^T\,\mathbf{e}_1\ -\ \bar{\mathcal{R}}_m\ \mathbf{y}_m\,\|_2 \quad .$$

Defining the vector $\mathbf{g}_{m+1}\ =\ \beta\,{\mathcal{Q}_{m+1}}^T\,\mathbf{e}_1$ and expanding $\|\,\mathbf{r}_m\,\|_2$ into two terms yields

$$\|\,\mathbf{r}_m\,\|_2\ =\ \|(\,\mathbf{g}_m\ -\ \mathcal{R}_m\ \mathbf{y}_m\,)\|_2\ +\ |g_{m+1}|^2\,. \tag{7}$$

Clearly, the minimum is achieved when

$$\mathbf{y}_m\ =\ \mathrm{argmin}_y\|\beta\,\mathbf{e}_1\ -\ \bar{\mathcal{H}}_m\ \mathbf{y}_m\,\|_2 \qquad \rightarrow \qquad \mathbf{y}_m\ =\ {\mathcal{R}_m}^{-1}\ \mathbf{g}_m\ =\ \beta\,{\mathcal{R}_m}^{-1}\ {\mathcal{Q}_{m+1}}^T\,\mathbf{e}_1\,. \tag{8}$$

The L_2 norm of the residual $\mathbf{r}_m$ is a commonly used measure of iterative convergence. An efficient means of calculating this norm is to combine equation (6) with equations (7) and (8)

$$\|\,\mathbf{r}_m\,\|_2\ =\ V_{m+1}\,(\beta\,\mathbf{e}_1\ -\ \bar{\mathcal{H}}_m\ \mathbf{y}_m\,)\ =\ |g_{m+1}|^2 \quad . \tag{9}$$

The quantity $|g_{m+1}|$ is a byproduct of the minimization problem solved in equation (8), requires no additional computations, and does not require decoding the solution vector $\mathbf{x}_m$. It is, however, sensitive to rounding errors. A combination of these two approaches is therefore used to monitor and terminate the GMRES iteration. Early stages of the iteration use the approximate residual $\|\,\mathbf{r}_m\,\|_2\ =\ |g_{m+1}|^2$, while in later stages (residual norm approaching machine precision, or two estimates significantly different) then $\mathbf{r}_m\ =\ \mathbf{b}\ -\ \mathcal{A}\ \mathbf{x}_m$ must be used to monitor the progress of the iteration. The pseudo-code for the GMRES(m) algorithm to solve the linear system $\mathcal{A}\ \mathbf{x}\ =\ \mathbf{b}$, is given in Algorithm 1.

Remark. For the sake of clarity our general algorithm described above is derived for unpreconditioned systems. However, the algorithm is also valid for preconditioned systems. In fact, the results described in section 5 are all for preconditioned systems.

Algorithm 1 (GMRES(m) to solve $\mathcal{A}\ \mathbf{x} = \mathbf{b}$.)

Choose $\mathbf{x}_0$ *and compute* $\mathbf{r}_0 = \mathbf{b} - \mathcal{A}\ \mathbf{x}_0$
Until convergence ***do:***
  Compute $\beta = \|\mathbf{r}_0\|_2$, $\mathbf{v}_1 = \mathbf{r}_0/\beta$ *and set* $j = 0$
  Until convergence and $j < m$ ***do:***
    $j = j + 1$
    $\mathbf{v}_{j+1} = \mathcal{A}\ \mathbf{v}_j$
    for $i = 1, \ldots, j$
      $\overline{\mathcal{H}}_{ij} = (\mathbf{v}_i, \mathbf{v}_{j+1})$
      $\mathbf{v}_{j+1} = \mathbf{v}_{j+1} - \overline{\mathcal{H}}_{ij}\mathbf{v}_i$
    end for
    $\mathbf{v}_{j+1} = \mathbf{v}_{j+1}/\overline{\mathcal{H}}_{j+1,\,j}$; $\overline{\mathcal{H}}_{j+1,\,j} = \left\|\mathbf{v}_{j+1}\right\|_2$
  end do
  Compute $\mathbf{y} = \min_{\mathbf{y}\in\Re_k} \left\|\beta\mathbf{e}_1 - \overline{\mathcal{H}}_{j+1,\,j}\, y\right\|$, $\mathbf{r}_j = V_{j+1}[\beta\mathbf{e}_1 - \overline{\mathcal{H}}_j\, \mathbf{y}]$ *and* $\mathbf{x}_j = \mathbf{x}_0 + V_j\, \mathbf{y}$
  If not converged set $\mathbf{x}_0 = \mathbf{x}_0 + \mathbf{x}_j$ *and* $\mathbf{r}_0 = \mathbf{r}_j$
end do

4 A Generalized Krylov Subspace Method

4.1 The Algorithm

A GMRES with enrichment algorithm denoted as ("GMRES-E") algorithm is developed; the algorithm is designed specifically to solve a variety of large, slowly varying linear systems of the form $\mathcal{A}^{\,i}\,\mathbf{x} = \mathbf{b}^{\,i}$. The kernel of the GMRES-E algorithm is essentially a conventional GMRES algorithm with k enrichment vectors $\xi_j,\ j = 1, 2, \cdots, k$ prepended to the test space, and is implemented as follows. Starting with the initial guess $\mathbf{x}_0$, the residual $\mathbf{r}_0{}^{\,i}$ is formed. Next, the vector $\mathbf{r}_0{}^{\,i}$ is orthogonalized with respect to the range η_j^{i-1} of the enrichment space ξ_j^{i-1} (η spans $\mathcal{A}\,\xi$) to obtain the orthogonal ($\perp$) component $\mathbf{r}_\perp{}^{\,i}$, which is then appended to the enriched space. The Arnoldi process is used to build a Krylov subspace based on the vector $\mathbf{r}_\perp{}^{\,i}$, then the current residual is minimized on that subspace. Note that any ξ is an admissible enrichment vector, which is a property that significantly extends the generality of GMRES-E. See figure 1 for a schematic of the algorithm.

Four critical subelements contribute to the generality or extend the efficiency of the GMRES-E algorithm. They are (see figure 1 for a schematic of the algorithm):

1. Determine an optimal starting solution $\bar{\mathbf{x}}_0$ based on previous solutions to the problems $\mathcal{A}^{\,i}\,\mathbf{x}^{\,i} = \mathbf{b}^{\,i}$. This subelement is applicable when $\mathcal{A}^{\,i}$ remains unchanged from previous one or more iterations (shown in green in figure 1).

2. Select a predetermined number k of enrichment vectors. This subelement is invoked at the beginning of each new problem in the sequence $[\cdots, i]$ or when the current problem is restarted. A new approach for choosing enrichment vectors is presented; this approach enhances the performance of the GMRES-E algorithm in cases where the accurate approximation of the eigenvectors is difficult (shown in blue in figure 1).

3. Construct the Arnoldi relationship $\mathcal{A}^{\,i}\, S_k \;=\; V_k\; \bar{\mathcal{H}}_k$ for the k enrichment vectors selected. Note that when $\mathcal{A}$ does not change, an existing Arnoldi relationship $\mathcal{A}^{\,i}\, S_m \;=\; V_{m+1}\; \mathcal{H}_m$ can be compressed or rotated into the desired form (shown in yellow in figure 1).

4. Use a Galerkin projection technique to preprocess the new $\mathbf{r}_0{}^{\,i}$ so that it is consistent with the currently available Arnoldi relation. This subelement ensures the admissibility of arbitrary enrichment vectors (shown in rose in figure 1).

We now present the theoretical underpinnings for the GMRES-E algorithm; we begin with the kernel and then include the four subelements. Note the similarity between the kernel and the conventional GMRES algorithm.

4.2 Kernel: GMRES with Enrichment

Let $\mathcal{A}$ be an $N \times N$ matrix and $\mathbf{b}$ a vector of length N; these define the linear system $\mathcal{A}\,\mathbf{x} = \mathbf{b}$ [see equation (1)]. Define or construct a kth-dimensional (nonorthogonal) vector space S_k and the corresponding orthogonal space V_k that satisfies the relation $\mathcal{A}\; S_k \;=\; V_k\; \mathcal{H}_k$. Next, append to V_k the orthogonal component of the current residual vector $\mathbf{r}_\perp$, and then, using the Arnoldi process, build the vector spaces S_m and V_{m+1} such that the expressions

$$\mathcal{A}\; S_m \quad = \quad V_m\; \mathcal{H}_m \;+\; \mathbf{v}_{m+1}\; h_{m+1,m}\, \mathbf{e}_m{}^T \quad = \quad V_{m+1}\; \bar{\mathcal{H}}_m \tag{10}$$

$$[\, V_{m+1}\,]^T\, V_{m+1} \;=\; \mathcal{I}_{m+1} \qquad ; \qquad \mathbf{r}_0{}^{\,i} \;=\; V_{m+1}\; \mathbf{f}_{m+1} \tag{11}$$

are satisfied. Note that the vector $\mathbf{f}_{m+1}$ is the projection of the residual $\mathbf{r}_0{}^{\,i}$ onto the space V_{m+1} . Using equations (10) and (11), GMRES-E builds an approximate solution to equation (1) of the form

$$\mathbf{x}_m \;=\; \mathbf{x}_0 \;+\; S_m\; \mathbf{y}_m \tag{12}$$

with the vector $\mathbf{y}_m \in \mathcal{R}^m$ and $\mathbf{y}_m$ determined such that the L_2 norm of the residual $\mathbf{r}_m = \mathbf{b} - \mathcal{A}\mathbf{x}_m$ is minimized over the vector space V_{m+1} . The approach that is used to minimize $\mathbf{r}_m$ in GMRES-E is identical to that in GMRES.

The residual $\mathbf{r}_m$ satisfies the expression $\mathbf{r}_m \;=\; \mathbf{r}_0 \;-\; V_{m+1}\; \bar{\mathcal{H}}_m\; \mathbf{y}_m$, which when simplified using equation (11) (and $\mathbf{f}_{m+1} \;=\; [\, V_{m+1}\,]^T\; \mathbf{r}_0$) yields

$$\mathbf{r}_m \;=\; \mathbf{r}_0 \;-\; V_{m+1}\; \bar{\mathcal{H}}_m\; \mathbf{y}_m \quad = \quad V_{m+1}\,(\, \mathbf{f}_{m+1} \;-\; \bar{\mathcal{H}}_m\; \mathbf{y}_m\,) \tag{13}$$

By construction, V_{m+1} is orthogonal, and the L_2 norm of $\mathbf{r}_m$ simplifies to $\|\, \mathbf{r}_m \,\|_2 \;=\; \|\, \mathbf{f}_{m+1} \;-\; \bar{\mathcal{H}}_m\; \mathbf{y}_m \,\|_2$. The residual $\|\, \mathbf{r}_m \,\|_2$ is minimized for a value of $\mathbf{y}_m$ such that $\mathbf{y}_m \;=\; \mathrm{argmin}_y \|\, \mathbf{f}_{m+1} \;-\; \bar{\mathcal{H}}_m\; \mathbf{y}_m \,\|_2$. Forming the $Q\,R$ decomposition of the matrix $\bar{\mathcal{H}}_m$ and substituting $\bar{\mathcal{H}}_m$ into the residual $\|\, \mathbf{r}_m \,\|_2$ produces

$$\|\, \mathbf{r}_m \,\|_2 \;=\; \|\, \mathcal{Q}_{m+1}{}^T\, \mathbf{f}_{m+1} \;-\; \bar{\mathcal{R}}_m\; \mathbf{y}_m \,\|_2 \quad .$$

Defining the vector $\mathbf{g}_{m+1} = \mathcal{Q}_{m+1}{}^T \mathbf{f}_{m+1}$, and expanding the residual as $\|\, \mathbf{r}_m \,\|^2 = \|(\, \mathbf{g}_m \;-\; \mathcal{R}_m\; \mathbf{y}_m \,)\|^2 + |g_{m+1}|^2$ implies that $\|\, \mathbf{r}_m \,\|_2$ is minimized by

$$\mathbf{y}_m \;=\; \mathrm{argmin}_y \|\, \mathbf{f}_{m+1} \;-\; \bar{\mathcal{H}}_m\; \mathbf{y}_m \,\|_2 \qquad \rightarrow \qquad \mathbf{y}_m \;=\; [\, \mathcal{R}_m\,]^{-1}\; \mathbf{g}_m \;=\; [\, \mathcal{R}_m\,]^{-1}\; \bar{\mathcal{Q}}_m{}^T\, \mathbf{f}_m \,. \tag{14}$$

Remark. The residual monitoring techniques that are used in GMRES are expandable to GMRES-E. An efficient (no cost) alternative to direct formation of $\mathbf{r}_m = \mathbf{b} - \mathcal{A}\ \mathbf{x}_m$ is also available with GMRES-E

$$\| \mathbf{r}_m \|^2 = \| \mathbf{f}_{m+1} - \bar{\mathcal{H}}_m\ \mathbf{y}_m \|^2 = \|(\mathbf{g}_m - \mathcal{R}_m\ \mathbf{y}_m)\|^2 + |g_{m+1}|^2 = |g_{m+1}|^2 \quad . \tag{15}$$

Remark. The matrix $\bar{\mathcal{H}}_m$ in GMRES-E is in general full, at least in the $k \times k$ submatrix that corresponds to the enrichment vectors. Householder rotations are used to form the $Q\ R$ decomposition in this case. (The fullness of the $k \times k$ submatrix depends on the particular choice of enrichment vectors.)

Remark. The GMRES-E algorithm generates the following vector spaces (with a slight abuse of the Krylov nomenclature $\mathcal{K}_m$):

$$\begin{aligned} \mathcal{K}_m(S) &= [\quad \xi_1, \quad \xi_2, \cdots, \quad \xi_k,\ \mathbf{r}_\perp\ , (\mathcal{P}\,\mathcal{A}\,)\,\mathbf{r}_\perp\ , (\mathcal{P}\,\mathcal{A}\,)^2\,\mathbf{r}_\perp\ , \cdots , (\mathcal{P}\,\mathcal{A}\,)^{m-k-1}\,\mathbf{r}_\perp\ ] \\ \mathcal{K}_{m+1}(V) &= [\,\mathcal{A}\,\xi_1,\ \mathcal{A}\,\xi_2, \cdots,\ \mathcal{A}\,\xi_k,\ \mathbf{r}_\perp\ , (\mathcal{P}\,\mathcal{A}\,)\,\mathbf{r}_\perp\ , (\mathcal{P}\,\mathcal{A}\,)^2\,\mathbf{r}_\perp\ , \cdots , (\mathcal{P}\,\mathcal{A}\,)^{m-k-1}\ \mathbf{r}_\perp\ , (\mathcal{P}\,\mathcal{A}\,)^{m-k}\,\mathbf{r}_\perp\ ] \end{aligned} \tag{16}$$

where $\mathcal{P}$ is the projection matrix $\mathcal{P} = (\mathcal{I} - \eta_k^{i-1}[\eta_k^{i-1}]^T)$, and is spanned by the basis vectors S_m and V_{m+1}. Appendix (A) contains a proof that the final $m-k$ vectors in $\mathcal{K}_m$ form a Krylov subspace.

The vectors S_m and V_{m+1} are closely related. Specifically, if we define a

$$V_{k+1 \to m} \equiv [\mathbf{v}_{k+1}, \cdots, \mathbf{v}_m]$$

then

$$S_m = [\, S_k\ ,\ V_{k+1 \to m}\,] \qquad ; \qquad V_{m+1} = [\, V_k\ ,\ V_{k+1 \to m}\,, \mathbf{v}_{m+1}] \tag{17}$$

which indicates that $\mathbf{S}_j = \mathbf{V}_j$ for $k+1 \le j \le m$. As such, both S_m and V_{m+1} do not need to be stored, only S_k and V_{m+1}.

4.2.1 Preprocessing Using Previous Solutions

Assume that $i-1$ problems have been solved with $\mathcal{A}$ unchanged. Preprocessing first projects $\mathbf{b}^{\,i}$ onto the orthogonal data space $span\{\, \mathbf{b}^{\,1}, \cdots, \mathbf{b}^{\,i-1}\}$ to determine $\mathbf{b}_\perp{}^{i}$, which is the orthogonal component of $\mathbf{b}^i$. The vector $\mathbf{b}^i$ is then decomposed into $\mathbf{b}^i = \mathbf{b}_\perp{}^{i} + (\mathbf{b}^i - \mathbf{b}_\perp{}^{i})$. The solution to the problem $\mathcal{A}\,\mathbf{x}^i = (\mathbf{b}^i - \mathbf{b}_\perp{}^{i})$ is already available from the solution space $span\{\mathbf{x}^1, \cdots, \mathbf{x}^{i-1}\}$. The problem that remains is $\mathcal{A}\ \mathbf{x}^{\,i} = \mathbf{b}_\perp{}^{i}$. After $\mathbf{x}^{\,i}$ is determined, $\mathbf{x}^{\,i}$ and the RHS vector $\mathbf{b}_\perp{}^{i}$ are appended to their respective subspaces.

In practical problems, instability can arise without the following modification to the preprocessing step [41]. Note that the equality that is satisfied in practice is $\mathcal{A}^i\mathbf{x}^i = \mathbf{b}^i - \mathbf{r}^i$, not $\mathcal{A}^i\mathbf{x}^i = \mathbf{b}^i$. The residual vector may be far from machine precision in magnitude. As such, orthogonalizing the new $\mathbf{b}^{\,i}$ against the data space $span\{\, \mathbf{b}^{\,1}, \cdots, \mathbf{b}^{\,i-1}\}$ can lead to instability. Thus, the appropriate data spaces to use in preprocessing are the solution space $span\{\, \mathbf{x}^{\,1}, \cdots, \mathbf{x}^{\,i-1}\}$ and the data space $span\{\, \mathbf{b}^{\,1} - \mathbf{r}^{\,1}, \cdots, \mathbf{b}^{\,i-1} - \mathbf{r}^{\,i-1}\}$.

Assume that preexisting solution data satisfies
$\mathcal{A}\ \mathbf{x}^{\,j} = \mathbf{b}^{\,j} - \mathbf{r}^{\,j} \quad ; \quad j = 1, \cdots, i-1$. Define the vector spaces

$$\Gamma^{i-1} = [\, \mathbf{x}^{\,1}, \cdots, \mathbf{x}^{\,i-1}] \quad ; \quad \Omega^{i-1} = [\, \mathbf{b}^{\,1} - \mathbf{r}^{\,1}, \cdots, \mathbf{b}^{\,i-1} - \mathbf{r}^{\,i-1}] \quad .$$

Orthogonalize Ω^{i-1} such that

$$\Omega^{i-1} \;=\; \Theta^{i-1}\,\mathcal{H}_{br} \quad ; \quad [[\Theta]^{i-1}]^T\Theta^{i-1} \;=\; \mathcal{I}^{i-1} \quad .$$

With these definitions, the data from all previous problems can be expressed as

$$\mathcal{A}\,\Gamma^{i-1} \;=\; \Theta^{i-1}\,\mathcal{H}_{br} \quad .$$

The preprocessing step is encapsulated in the following lemma.

Lemma 1 *The L_2 optimal starting guess $\mathbf{x}_0{}^i$ that is constructed from the space Γ^{i-1}, and the initial guess $\bar{\mathbf{x}}_0{}^i$ is given by*

$$\mathbf{x}_0{}^i \quad = \quad \bar{\mathbf{x}}_0{}^i \;+\; \Gamma^{i-1}\mathcal{H}_{br}^{-1}[\Theta^{i-1}]^T\,\bar{\mathbf{r}}_0$$

The starting residual is

$$\mathbf{r}_0 \quad = \quad (\mathcal{I} \;-\; \Theta^{i-1}[\Theta^{i-1}]^T)\,\bar{\mathbf{r}}_0$$

Proof : The general expression for $\mathbf{x}_0$, formed from an update from the space Γ^{i-1} with starting guess $\bar{\mathbf{x}}_0$, is

$$\mathbf{x}_0 \quad = \quad \bar{\mathbf{x}}_0 \;+\; \Gamma^{i-1}\mathbf{y}^{i-1} \tag{18}$$

with $\mathbf{y}^{i-1}$ an unknown $i-1$-vector. The general residual $\mathbf{r}_0$ is then given by

$$\mathbf{r}_0 \;=\; \bar{\mathbf{r}}_0 \;-\; \Theta^{i-1}\mathcal{H}_{br}\mathbf{y}^{i-1} \;=\; (\mathcal{I} \;-\; \Theta^{i-1}[\Theta^{i-1}]^T)\,\bar{\mathbf{r}}_0 \;+\; \Theta^{i-1}\{\,[\Theta^{i-1}]^T\,\bar{\mathbf{r}}_0 \;-\; \mathcal{H}_{br}\mathbf{y}^{i-1}\,\} \tag{19}$$

Solving $\{\,[\Theta^{i-1}]^T\,\bar{\mathbf{r}}_0 \;-\; \mathcal{H}_{br}\mathbf{y}^{i-1}\,\}$ for $\mathbf{y}^{i-1}$ yields

$$\mathbf{y}^{i-1} \;=\; \mathcal{H}_{br}^{-1}\,[\Theta^{i-1}]^T\,\bar{\mathbf{r}}_0$$

and the expression for $\mathbf{r}_0$ follows immediately. □

Remark. Note that unlike the minimization step in GMRES and GMRES-E, the equation $\{[\Theta^{i-1}]^T\bar{\mathbf{r}}_0 - \mathcal{H}_{br}\mathbf{y}^{i-1}\}$ can be solved identically for $\mathbf{y}^{i-1}$, instead of resorting to an L_2 minimization technique.

4.2.2 Selecting Enrichment Vectors

The choice of enrichment vectors is a delicate task that has received considerable past attention in the literature. Indeed, the selection algorithm has a profound impact on the overall performance of the enrichment algorithms. Although at least three well-established methodologies exist for selecting enrichment vectors, we include a fourth and a fifth.

I The first approach (e.g., see references [24–27,42,43]) identifies eigenvalues (and eigenvectors) that are inadequately clustered by the preconditioner and, thus, contribute to poor convergence. Ritz pairs [27, 44] (e.g., Ritz-values and vectors) are commonly used to approximate the eigenvalues and eigenvectors of the matrix $\mathcal{A}$. They are obtained by choosing vector $\breve{\xi}_m$ such that the error is minimized in the relation $[\,\mathcal{A} \;-\; \breve{\theta}\mathcal{I}]\,S_m\;\breve{\xi}_m \;=\; 0$. Two popular techniques for minimizing the error in the Ritz problem are the Galerkin Projection (GP) and the Petrov-Galerkin projection. The Galerkin and Petrov-Galerkin projections produce the *conventional* Ritz and *harmonic* Ritz pairs, respectively. See Appendix B for derivations of the harmonic and conventional Ritz formulations. One or more Ritz vectors are used to enrich the next S_m and converge to the exact eigenvectors as the iteration progresses.

II A second approach (e.g., see references [45–54]) uses physical arguments to construct approximate eigenvectors, a technique that is highly effective for problems with a recognizable physical attribute that contributes to problematic eigenvalues. Much research, including the early work of Nicolaides [45], demonstrates the use of algebraic deflation vectors to deflate coarse-grid information. Indeed, using piecewise constant or piecewise linear (see Verkaik et al. [52]) deflation vectors is often sufficient. Vuik et al. [47, 48, 55] used this technique to accelerate the convergence of a preconditioned CG method that was used to solve the time-dependent diffusion equation with discontinuous coefficients. Furthermore, they showed (through analysis and numerical experiments) that perturbations in the approximate eigenvectors had limited influence on the efficacy of eigenvector enrichment in this context. This approach has been shown to be effective on the more complicated cases that involve time-dependent problems with moving discontinuities [54].

III A third approach (e.g., see reference [42]) uses vectors that correspond to the eigenvector that significantly impacted the convergence of past iterations. This approach asks the questions, "How much worse would the convergence be at step m if the iteration was restarted at step s ($s < m$) and $s - m$ vectors were discarded?" and "Furthermore, which vectors should be retained?" To answer these questions, optimal solutions are obtained via projections onto the full and truncated spaces. The residuals are compared in each case, and the truncated modes are identified and ranked in importance.

IV A fourth approach is to use the Ritz vectors from a previous and slightly different $\mathcal{A}$. This step is only warranted if $\mathcal{A}^{i-1} \approx \mathcal{A}^{i}$. The Ritz vectors are computed in a manner that is similar to that described in approach I, while the Arnoldi relation is constructed as described in II.

V A fifth approach is to retain the entire space $\mathcal{K}_m$ and the accompanying Arnoldi relation (with the assumption that memory resources are sufficient). This approach is motivated by cases for which accurate approximations of the eigenvectors are not forthcoming. This is frequently the case when the desired tolerance for the linear system is reached well before the Ritz vectors are accurate approximations of the problematic eigenvectors. This approach could be considered a trivial extension of the first (or the third) approach in the limit where all m harmonic Ritz vectors are retained rather than only a few.

The preferred approach for choosing enrichment vectors in GMRES-E is the first, which is closely patterned after that of Morgan [26]. The generalized eigenvalue problem derived in Appendix B [eq. (41)] is solved to determine m harmonic Ritz pairs. Next, the selection algorithm in section B.2 chooses the worst k eigenvectors and loads them into an $m \times p$ matrix $\bar{\mathcal{P}}_k$. Right-multiplying S_m yields the desired Ritz space $S_k = S_m \bar{\mathcal{P}}_k$. Enrichment by this approach is referred to as *eigenvector enrichment* and is denoted in the results section by the subscript E.

The fifth approach is the preferred alternative if accurate approximations of the problematic eigenvectors are not forthcoming. (This is commonly the case in defect-correction iterations and is the topic of a future paper.) Note that this approach is equivalent to choosing $\bar{\mathcal{P}}_k$ to be $\mathcal{I}$, with $k = m$. Enrichment by this approach is referred to as *appending enrichment*, and is denoted in the results section by the subscript A.

4.2.3 Data Rotation and Compression

In preparation for the next cycle (i.e., restart or a new problem), the existing Arnoldi relation must be rotated or compressed. If we assume that important information is stored in the matrix $\bar{\mathcal{P}}_k$, then this step is accomplished as follows.

Right-multiply the Arnoldi relation $\mathcal{A}^{\,i-1} S_n = V_{n+1}\, \bar{\mathcal{H}}_n$ by the matrix $\bar{\mathcal{P}}_k$ to produce the expression

$$\mathcal{A}\; S_n\; \bar{\mathcal{P}}_k \quad = \quad V_{n+1}\; \bar{\mathcal{H}}_n\; \bar{\mathcal{P}}_k \quad . \tag{20}$$

Note that this step is only possible when $\mathcal{A}^{\,i-1} = \mathcal{A}^{\,i}$ and when an Arnoldi relationship is available. When $\mathcal{A}^{\,i-1} \neq \mathcal{A}^{\,i}$ or when the Arnoldi relationship is not available, data rotation and compression cannot be used. In this case, the selected enrichment vectors must be left-multiplied with $\mathcal{A}$ to construct the Arnoldi relationship $\mathcal{A}^{\,i} S_k = V_k\; \mathcal{H}_k$.

Data compression can proceed in two similar ways, which differ only in how the nonsquare matrix $\bar{\mathcal{H}}_n\, \bar{\mathcal{P}}_k$ is decomposed. The first technique uses $Q\,R$ factorizations to decompose the matrix product $\bar{\mathcal{H}}_n\; \bar{\mathcal{P}}_k$. This technique is computationally efficient, although numerical experiments reveal that it is more susceptible to roundoff errors. The second technique uses a singular value decomposition to decompose $\bar{\mathcal{H}}_n\; \bar{\mathcal{P}}_k$. This technique has better stability properties but is more computationally intensive.

Q R Compression:
The $Q\; R$ technique begins by substituting $\bar{\mathcal{H}}_n\; \bar{\mathcal{P}}_k = \bar{\mathcal{Q}}_n\; \mathcal{R}_k$ into equation (20) to yield the expression

$$\mathcal{A}\; S_n\; \bar{\mathcal{P}}_k \quad = \quad V_{n+1}\; \bar{\mathcal{H}}_n\; \bar{\mathcal{P}}_k \quad = \quad V_{n+1}\; \bar{\mathcal{Q}}_n\; \mathcal{R}_k \quad . \tag{21}$$

Next, we define

$$S_k = S_n\; \bar{\mathcal{P}}_k \quad ; \quad V_k = V_{n+1}\; \bar{\mathcal{Q}}_n \tag{22}$$

With these definitions, the compressed system becomes

$$\mathcal{A}\; S_k \quad = \quad V_{n+1}\; \bar{\mathcal{Q}}_n\; \mathcal{R}_k \quad = \quad V_k\; \mathcal{R}_k \tag{23}$$

and retains the following property.

Lemma 2 *The compressed matrix V_k is orthogonal.*

Proof :

$$V_k{}^T\, V_k \;=\; \bar{\mathcal{Q}}_n{}^T\, V_{n+1}{}^T\, V_{n+1}\; \bar{\mathcal{Q}}_n \;=\; \mathcal{I}_k$$

□

S V D Compression:
The singular value decomposition ($S\,V\,D$) technique begins by substituting $\bar{\mathcal{H}}_n\; \bar{\mathcal{P}}_k = \bar{\mathcal{Q}}_n\; \mathcal{D}_k\; \mathcal{W}_k^T$ into equation (20) to yield the expression

$$\mathcal{A}\; S_n\; \bar{\mathcal{P}}_k \quad = \quad V_{n+1}\; \bar{\mathcal{H}}_n\; \bar{\mathcal{P}}_k \quad = \quad V_{n+1}\; \bar{\mathcal{Q}}_n\; \mathcal{D}_k\; \mathcal{W}_k^T \tag{24}$$

Right-multiplying by $[\,\mathcal{W}_k^T\,]^{-1} = \mathcal{W}_k$ and using the definitions

$$S_k = S_n\; \bar{\mathcal{P}}_k\; \mathcal{W}_k \quad ; \quad V_k = V_{n+1}\; \bar{\mathcal{Q}}_n\; \mathcal{D}_k \tag{25}$$

yields the compressed system

$$\mathcal{A}\ S_k \quad = \quad V_k\ \mathcal{D}_k \tag{26}$$

Inspection reveals that this form also retains the desired orthogonality property for the compressed matrix V_k .

Remark. The $Q\ R$ compression technique is significantly more efficient when $\bar{\mathcal{P}}_k \ = \ \mathcal{I}_n$. In this case S_n remains unchanged; thus the work is reduced by n^2 matrix-vector products (i.e. n^2 calls to the blas1 routine DAXPY).

Remark. Both new Arnoldi relations [eqs. (23) and (26)] can be expressed symbolically in the form $\mathcal{A}\ S_k \ = \ V_k\ \mathcal{R}_k$ where $\mathcal{R}_k$ is an upper triangular matrix.

4.2.4 Initialization by Galerkin Projection

A mechanism is now needed to include the current residual in the subspace and start the next iteration. The Galerkin projection (GP) technique [1, 26, 56, 57] provides such a mechanism. Furthermore, this technique is L_2 optimal.

The fundamental steps of the GP are expressed in the following lemma.

Lemma 3 *Assume that the relation* $[\ \mathcal{A}\ S_k \ = \ V_k\ \mathcal{R}_k\ ]$ *has been constructed. The* L_2 *optimal starting solution* $\mathbf{x}_0$ *, which is constructed from vectors in* S_k *and any starting guess* $\bar{\mathbf{x}}_0$ *, is*

$$\mathbf{x}_0 \quad = \quad \bar{\mathbf{x}}_0 \ + \ S_k\ \mathcal{R}_k{}^{-1}\ V_k{}^T\, \bar{\mathbf{r}}_0$$

and the starting residual is

$$\mathbf{r}_0 \quad = \quad (\mathcal{I} \ - \ V_k\ V_k{}^T)\, \bar{\mathbf{r}}_0$$

Proof : The general expression for $\mathbf{x}_0$, formed from an update from the space S_k with starting guess $\bar{\mathbf{x}}_0$, is

$$\mathbf{x}_0 \quad = \quad \bar{\mathbf{x}}_0 \ + \ S_k\ \mathbf{y}_k \tag{27}$$

where $\mathbf{y}_k$ is an unknown k-vector. Using $\mathcal{A}\ S_k \ = \ V_k\ \mathcal{R}_k$, the general residual $\mathbf{r}_0$ is then given by

$$\mathbf{r}_0 \quad = \quad \bar{\mathbf{r}}_0 \ - \ V_k\ \mathcal{R}_k\ \mathbf{y}_k \tag{28}$$

Adding and subtracting $V_k\ V_k{}^T\, \bar{\mathbf{r}}_0$ yields the expression

$$\mathbf{r}_0 \quad = \quad (\ \mathcal{I}_n \ - \ V_k\ V_k{}^T)\, \bar{\mathbf{r}}_0 \ + \ V_k\, (\ V_k{}^T\, \bar{\mathbf{r}}_0 \ - \ \mathcal{R}_k\ \mathbf{y}_k\ ) \quad . \tag{29}$$

The term $(\ \mathcal{I}_n \ - \ V_k\ V_k{}^T)\, \bar{\mathbf{r}}_0$ is the component of $\bar{\mathbf{r}}_0$ that is orthogonal to the space V_k , and it is independent of the choice of $\mathbf{y}_k$. Solving the second term in equation (29) implies that the residual $\| \mathbf{r}_0 \|_2$ is minimized for the value of $\mathbf{y}_k$

$$\mathbf{y}_k \ = \ \mathrm{argmin}_y \| V_k{}^T\ \bar{\mathbf{r}}_0 \ - \ \mathcal{R}_k\ \mathbf{y}_k \|_2 \qquad \rightarrow \qquad \mathbf{y}_k \ = \ [\ \mathcal{R}_k\ ]^{-1}\ V_k{}^T\ \bar{\mathbf{r}}_0 \qquad . \tag{30}$$

Substituting equation (30) into equations (27) and (28) yields the desired result. □

Remark. Three distinct restart scenarios exist: 1) restart an existing problem $\mathcal{A}\ \mathbf{x} \ = \ \mathbf{b}$ (e.g., insufficient memory), 2) solve the same system of equations with a different or multiple RHS $\mathbf{b}$ (e.g., $\mathcal{A}\ \mathbf{x} \ = \ \tilde{\mathbf{b}}$), and 3) solve a different but closely related problem $\tilde{\mathcal{A}}\ \mathbf{x} \ = \ \tilde{\mathbf{b}}$. In the first two restart scenarios which involve constant $\mathcal{A}$, the relation $\mathcal{A}\ S_k \ = \ V_k\ \mathcal{H}_k$ is available from the

previous problem $\mathcal{A}^{i-1}\,\mathbf{x}^{i-1} = \mathbf{b}^{i-1}$. The third restart scenario requires the construction of a new Arnoldi relation given the starting vectors S_k.

As a final step of the initialization, normalize the optimal residual $\mathbf{r}_0$ and create the starting vector

$$\mathbf{v}_{k+1} = \frac{\mathbf{r}_0}{\| \mathbf{r}_0 \|_2} \tag{31}$$

Appending $\mathbf{v}_{k+1}$ to V_k in the relation $\mathcal{A}\ S_k = V_k\ \mathcal{R}_k$ yields

$$\mathcal{A}\ S_k \quad = \quad [\, V_k \mid \mathbf{v}_{k+1} \,] \begin{bmatrix} \mathcal{R}_k \\ \mathcal{Z}_k \end{bmatrix} \quad = \quad V_{k+1}\ \bar{\mathcal{R}}_k \tag{32}$$

with $\mathcal{Z}_k = [0, \cdots, 0]$. The matrix $\bar{\mathcal{R}}_k$ is of dimension $k+1 \times k$, and row $k+1$ is entirely zero.

Lemma 4 *The matrix V_{k+1} is orthogonal.*

Proof :

$$V_k{}^T\, V_k = \mathcal{I}_k \quad ; \quad \mathbf{v}_{k+1}{}^T\, V_k = \| \mathbf{r}_0 \|_2^{-1}\, \bar{\mathbf{r}}_0{}^T (\mathcal{I} - V_k\ V_k{}^T)\, V_k = 0 \quad ; \quad \mathbf{v}_{k+1}{}^T\, \mathbf{v}_{k+1} = 1 \tag{33}$$

□ The system $\mathcal{A}\, S_k = V_{k+1}\, \bar{\mathcal{R}}_k$ is now ready for the generation of an $m-k$ dimensional Krylov space with $\mathbf{v}_{k+1}$ as its starting vector. Pseudo-code for the GMRES(m) algorithm enriched with q vectors, and designed to solve a sequence of linear systems $\mathcal{A}^{\kappa}\,\mathbf{x} = \mathbf{b}^{\kappa}$ is given in Algorithm 2.

5 Numerical Experiments

5.1 Test Problems

5.1.1 Advection-Diffusion

The first test problem solves the *steady* advection-diffusion equation

$$\begin{array}{rcl} \epsilon_x \frac{\partial}{\partial x}\frac{\partial u}{\partial x} + \epsilon_y \frac{\partial}{\partial y}\frac{\partial u}{\partial y} + \epsilon_z \frac{\partial}{\partial z}\frac{\partial u}{\partial z} & = & D\frac{\partial u}{\partial x} + E\frac{\partial u}{\partial y} + F\frac{\partial u}{\partial z} \quad ; \quad 0 \le x, y, z \le 1 \\ \alpha \frac{\partial u}{\partial n} + \beta u & = & \gamma \end{array} \tag{34}$$

with the parameters α, β, and γ set to $\mathcal{O}(1)$ constants.

The diffusion and advection parameters are assigned to be $\epsilon_x = \epsilon_y = \epsilon_z = \epsilon$, and

$$D(x,y,z) = \exp[(+x)y] \quad ; \quad E(x,y,z) = \exp[(-x)y]\sin(\pi z) \quad ; \quad F(x,y,z) = \exp[(-x)y]\sin(-\pi z)$$

Three values are used for parameters ϵ in this study: $\epsilon = 1, \frac{1}{10}, \frac{1}{100}$. The problem is discretized on a single $[\ (N_x, N_y, N_z) = (141, 99, 79)\ ]$ grid, and the Jacobian of the discretization operator produces the matrix $\mathcal{A}$. In principle, the steady solution is obtained by solving a single system $\mathcal{A}\ \mathbf{x} = \mathbf{b}$, where the vector $\mathbf{b}$ represents the initial solution field. Herein, to approximately emulate the conditions that are encountered in unsteady simulations, multiple RHS vectors $\mathbf{b}$, $[\ \mathbf{b}^{\,i} = \mathbf{1} + 0.1\, rand()\ ]$, $i = 0, 6$, are solved sequentially as $\mathcal{A}\ \mathbf{x} = b^i$. The $ILU(k)$ (ILU with level k of fill-in) class of preconditioners with $0 \le k \le 4$, which are formed from the exact matrix $\mathcal{A}$, are used exclusively in this study. This problem is characterized by strongly varying advection terms and illustrates the complementary nature of enrichment for $ILU(k)$ preconditioners.

Algorithm 2 (GMRES(m) with q enrichment vectors to solve $\mathcal{A}^{\kappa}\mathbf{x} = \mathbf{b}^{\kappa}$.)

Until nonlinear convergence ***do:***
  $\kappa = \kappa + 1$
  Construct $\mathcal{A}^{\kappa}$ *and* $\mathbf{b}^{\kappa}$
  Choose $\overline{\mathbf{x}}_0$ *and compute* $\overline{\mathbf{r}}_0 = \mathbf{b} - \mathcal{A}^{\kappa}\overline{\mathbf{x}}_0$
  If $\mathcal{A}^{\tau} = \cdots = \mathcal{A}^{\kappa-1} = \mathcal{A}^{\kappa}$,
    Project $\overline{\mathbf{x}}_0$ *and* $\overline{\mathbf{r}}_0$ *onto* $\mathcal{A}\,\mathbf{x}^{[\tau,\cdots,\kappa-1]} = \mathbf{b}^{[\tau,\cdots,\kappa-1]}$ *for optimal initial* $\mathbf{x}_0$, $\mathbf{r}_0$
  Until linear convergence ***do:***
    Initialize S_q^{κ} *with* q *enrichment vectors*
      If *restart: problematic eigenvectors from* $\mathcal{A}^{\kappa}$
      If *same* $\mathcal{A}$, *new* $\mathbf{b}^{\kappa}$, *and available memory:* $S_q^{\kappa} = S_m^{\kappa-1}$
    Construct the square Arnoldi relation $A^{\kappa}S_q = V_q H_q$
    Orthogonalize $\mathbf{r}_0$ *against* V_q *such that* $\tilde{\mathbf{r}}_0 \perp V_q$
    Compute $\beta = \|\tilde{\mathbf{r}}_0\|_2$; $\mathbf{v}_{q+1} = \tilde{\mathbf{r}}_0/\beta$; $\overline{H}_{q+1,j} = \underline{0}^T$, $j = 1, q$; *Set* $j = q+1$
    Until convergence and $j < m$ ***do:***
      $j = j + 1$
      $\mathbf{v}_{j+1} = \mathcal{A}^{\kappa}\mathbf{v}_j$
      for $i = 1, \ldots, j$
        $\overline{H}_{ij} = (\mathbf{v}_i, \mathbf{v}_{j+1})$
        $\mathbf{v}_{j+1} = \mathbf{v}_{j+1} - \overline{H}_{ij}\mathbf{v}_i$
      end for
      $\mathbf{v}_{j+1} = \mathbf{v}_{j+1}/\overline{H}_{j+1,j}$; $\overline{H}_{j+1,j} = \left\|\mathbf{v}_{j+1}\right\|_2$
    end do
    Define $S = [S_q, V_{q+1\to j}]$; $\mathbf{x}_j = \mathbf{x}_0 + S\,y$
    Compute $\mathbf{y} = \min_{\mathbf{y}\in\Re_q} \left\|\beta\mathbf{e}_q - \overline{H}_j\,y\right\|$; $\mathbf{r}_j = V_{j+1}[\beta\mathbf{e}_q - \overline{H}_j\,y]$
    Estimate problematic eigenvectors from $A^{\kappa}S_j = V_{j+1}\overline{H}_j$
    If *linear converged*
      Save $\mathbf{x}^{\kappa}$ *and* $\mathbf{b}^{\kappa}$ *as* $\mathcal{A}\,\mathbf{x}^{[\tau,\cdots,\kappa]} = \mathbf{b}^{[\tau,\cdots,\kappa]}$
      Exit to nonlinear loop
    else
      Restart : $\mathbf{x}_0 = \mathbf{x}_0 + \mathbf{x}_j$ *and* $\mathbf{r}_0 = \mathbf{r}_j$
    endif
  end do
end do

5.1.2 Diffusion

The second problem is purely diffusive and assigns the advection parameters to $D = E = F = 0$. The effect of (grid) anisotropy is studied by assigning the diffusion coefficients ϵ_x, ϵ_y, ϵ_z to be

$$\begin{array}{rcrcrcl} \epsilon_x(x,y,z) & = & \epsilon_y(x,y,z) & = & \epsilon_z(x,y,z) & = & 1 \\ 100\epsilon_x(x,y,z) & = & 10\epsilon_y(x,y,z) & = & \epsilon_z(x,y,z) & = & 1 \\ 10000\epsilon_x(x,y,z) & = & 100\epsilon_y(x,y,z) & = & \epsilon_z(x,y,z) & = & 1 \quad . \end{array}$$

The problem is discretized on a family of four grids of densities $33 \times 33 \times 33$, $65 \times 65 \times 65$, $129 \times 129 \times 129$ and $257 \times 257 \times 257$, respectively. All cases are solved with $\mathbf{b} = [1, \cdots, 1]^T$. The preconditioners used for this study are $ILU(k)$ as well as "multiple-sweep Jacobi" [Jac(k)]. The study addresses the efficacy of enrichment for cases with thousands of closely related problematic eigenmodes.

All terms in both problems are discretized on a uniform mesh using second-order finite-difference stencils. The resulting matrix $\mathcal{A}$ has seven non-zero diagonals in 3D. The subroutine used to generate the fully discrete matrix $\mathcal{A}$ corresponding to equation (34) is from SPARSKIT, Version 2 [58].

5.1.3 2D Unsteady: Hexstream

The second test problem is a fully turbulent, unsteady flow around a 25-percent thick wind turbine profile that is solved with the Hexstream [14, 15] software. A sequence of Jacobians and right-hand sides are generated that correspond to the three nonlinear solutions that are required to advance one physical time step with a three-stage implicit RK scheme. The time step in this numerical experiment occurs after the completion of five vortex shedding cycles. The total number of linear equations that are solved at each stage are 6, 3, and 4, respectively, for a total of 13 independent linear solutions. The Jacobian matrix is frozen for all required linear solutions within each stage. The Jacobians are constructed with a perturbation method but contain only the nearest neighbor contributions. See Figures 6 and 7 for the wind turbine grid and a temporal solution of the turbulent viscosity.

The Hexstream flow solver is based on the Reynolds averaged Navier-Stokes (RANS) equations, which describe conservation of mass, momentum, and energy. The one-equation turbulence model of Spalart and Allmaras [59] is used as a closure model. The space discretization is performed with the cell-centered, conservative, finite-volume scheme. The convective flux is discretized by using a second-order central scheme with a Jameson type of scalar artificial dissipation. Finally, the integration in time is performed with a third-order ESDIRK scheme [60, 61].

5.2 Computational Considerations

One goal of this study is to establish the effectiveness of the enrichment GMRES approach in environments that are routinely encountered by practitioners in solving large linear systems ($> 10^6$ unknowns). Unfortunately, effective preconditioners are not always available; thus, determining the effectiveness of enrichment by using both strong and weak preconditioners is desirable. The $ILU(k)$ framework provides a simple means for building variable preconditioners. Changing the level of fill-in that is allowed in the L and U factors substantially changes the characteristics of the preconditioner. The $ILU(k)$ preconditioners are derived from the SPARSKIT [58] subroutine *ILUK*.

The CPU time that is required to form the incomplete LU decomposition is not included in the total simulation times. Neglecting the factorization times favors high-quality approximate inverses. This decision, however, reflects our interest in steady-state and time-dependent simulations for which the Jacobian matrix $\mathcal{A}$ is approximately factored, stored, and reused many times (> 10) before refactorization is necessary, which mitigates the impact of costly factorizations.

Simulations that use Ritz vectors for enrichment, choose the vectors by first assigning a merit to each Ritz value and then choosing the k Ritz values with the smallest merit. (See Appendix B for a derivation of the conventional and harmonic Ritz procedures.) Complex eigenvalues are

accommodated by using two Ritz vectors, one each for the real and imaginary components. If the Ritz problem does not yield exactly k eigenvectors, then $k-1$ vectors are used.

6 Results

6.1 Advection-Diffusion

6.1.1 Preliminaries

In tables 1, 2, and 3 we compare the impact of the enrichment on the linear advection-diffusion test problem for values of the parameter $\epsilon = 1, \frac{1}{10}, \frac{1}{100}$, respectively. Presented is the CPU time (i.e., the number of seconds on a single processor) that were required to reach a specific residual tolerance, as a function of 1) the enrichment vectors, 2) the Krylov subspace vectors, 3) the preconditioners, and 4) the enrichment-vector initialization. The first column in tables 1, 2, and 3 indicates the number of enrichment vectors (0, 2, 4, 8, 16, 24) that were used for the simulation; the first row after the "preconditioner" label indicates the number of Krylov vectors (30, 40, 50, 60, 70) that were used between restarts. The two preconditioners that were used in the study are $ILU(0)$, and $ILU(2)$. A residual reduction of ten orders (10^{-10}) was the criteria that was used to stop the simulations. Numbers that are less than or equal to zero denote the logarithm of the final residual, which is given for simulations that failed to reach the specified tolerance after 2000 iterations for any of the RHS vectors $\mathbf{b}_j$.

The two selection algorithms that were used in this study to build the enrichment space are the numbers I and IV which are given in section 4.2.2. Number V was not considered because of insufficient storage.

6.1.2 Eigenvector Enrichment

The positive impact of enrichment is most evident in the $\epsilon = \frac{1}{10}$ study. All cases stall for Krylov subspace dimensions (≤ 70) when two or fewer eigenvectors are used to enrich the subspace. Slow convergence is observed for a dimension of $60-70$ and four enrichment vectors, with the exception of the ineffective $ILU(0)$ preconditioner. The use of eight enrichment vectors proves effective for nearly all combinations of parameters. Similar trends are observed in the $\epsilon = 1$ study, although the tendency to stall is less likely. The $\epsilon = \frac{1}{100}$ case shows acceptable convergence in most cases.

Although the effect of enrichment is dependent on ϵ in this study, efficiency generally increases with increasing enrichment up to approximately $8-16$ vectors, followed by a slight decline. Convergence times with enrichment are two to four times smaller than those without enrichment. Enrichment has a greater impact on those cases that have difficulty converging [e.g., weak preconditioners like $ILU(0)$].

Although large dimensional Krylov subspaces (e.g., 70) usually provide faster convergence, this effect is of secondary importance relative to that of enrichment. For example, large Krylov spaces without enrichment are far less efficient than small spaces with $8-16$ enrichment vectors. Thus, in this example eliminating the problematic eigenvectors via enrichment is easier than building a large Krylov subspace.

The convergence trends for each value of the parameter ϵ can be predicted by the number (and location) of the eigenvalues of $\mathcal{A}\,M^{-1}$. To illustrate, figures 2, 3, 4, and 5 present an eigenvalue study. Figure 2 shows the first 60 Ritz values (compared with 1.1 Million eigenvalues) of the matrix

$\mathcal{A}\,M^{-1}$, with the matrix M derived from an $ILU(0)$ or an $ILU(2)$ preconditioner. Ritz values are generally not eigenvalues, but in this case the error in the Ritz values near the origin is small due to repeated Krylov subspace enrichment. Indeed the Ritz value error as determined using equation (46) indicates three to six significant digits. Although it appears that the Ritz values are converging to eigenvalues, it cannot be determined whether all of the eigenvalues near the origin are being identified.

Figure 3 shows an expanded view of the eigenvalues near the complex origin (0,0), using $ILU(1)$ with four different enrichment scenarios to approximate the eigenvalues. The dimension of the Krylov subspace is 60 in all four cases, while the number of enrichment vectors is $10, 20, 30, 40$, respectively. The vectors are approximated using Harmonic Ritz vectors corresponding to the k Ritz values closest to the origin. Note that each enriched eigenspace is (almost always) a proper subset of all larger spaces, and that the new eigenvalues in each successive space move progressively further from the origin. The enrichment procedure is identifying all the important eigenvalues near the origin.

Figures 4 and 5 show the eigenvalues of $\mathcal{A}\,M^{-1}$ located near the origin for the $\epsilon = \frac{1}{10}$ and $\epsilon = \frac{1}{100}$ test cases. Both the $ILU(0)$ and $ILU(2)$ preconditioners (M) are used in the study. Recall that the iteration for the $\epsilon = \frac{1}{10}$ test case was stalled for both preconditioners until approximately 10 eigenvalues were used to enrich the space, while the $\epsilon = \frac{1}{100}$ test case converged even without enrichment. Careful examination of the eigenvalues for each case reveals the reason for this behavior. First, note that numerous eigenvalues (e.g., 10) are outside the unit circle in the $\epsilon = \frac{1}{10}$ test case. (At this resolution, the unit circle is the vertical axis at the origin. Thus, eigenvalues in the LHP are outside the conventional stability region.) These eigenvectors, if not fully resolved by the restarted GMRES algorithm, could stall the iteration. The $\epsilon = \frac{1}{100}$ case has only one LHP eigenvalue. Evidently, for these test cases, the restarted GMRES algorithm is capable of isolating a single $\epsilon = \frac{1}{100}$ eigenvector but incapable of isolating all of the spurious modes in the $\epsilon = \frac{1}{10}$ case without the assistance of enrichment. Second, note that while the $ILU(2)$ preconditioner results in spurious eigenvalues that extend further into the LHP than those produced by the $ILU(0)$ preconditioner, it clusters most of the eigenvalues away from the origin. The $ILU(2)$ preconditioner outperformed $ILU(0)$ for all values of the parameter ϵ. Again this result is consistent with the notion that the restarted GMRES algorithm is effective at isolating a few spurious eigenvalues but has trouble with numerous unclustered eigenvalues.

6.1.3 Initialization

Appropriate initialization of the enrichment vectors can also significantly impact the overall convergence rate. The impact of reusing Ritz vectors from previous problems (section 4.2.2, element IV) is shown in tables 1, 2 and 3. Columns two through six show a conventional enrichment study, for which the starting vector is $\mathbf{x}_0 = 0$. Consequently, several Krylov Arnoldi cycles are necessary to accurately represent the enrichment vectors. Columns 7 - 11 show an identical study, except that the enrichment vectors are initialized using accurate data obtained from a previous study. (The study has the same matrix $\mathcal{A}$ but a different RHS vector $\mathbf{b}$.)

Table 1 clearly shows that reusing the Ritz vectors from a previous problem is very effective for the case $\epsilon = 1$. This is especially true when 16 and 24 enrichment vectors are used; a large speedup (i.e., factor of 2 to 3) is realized. The Galerkin projection initialization is much less effective for the $\epsilon = \frac{1}{10}$ and $\epsilon = \frac{1}{100}$ cases. In fact, in some cases, the Galerkin projection initialization slows down

the convergence. Either matrix asymmetry or indefiniteness (or both) could account for this trend. Evidently, more work is needed to quantify the influences on the effectiveness of initialization.

Table 4 displays the effect of the preprocessing step (i.e., projection initialization) on the starting residual in the advection-diffusion test problem using a diffusion parameter $\epsilon = \frac{1}{10}$. The preprocessing step first assumes $\bar{\mathbf{x}}_0 = 0$, which yields a temporary starting residual $\bar{\mathbf{r}}_0$. This residual is then projected onto the available solution space, thus producing a new residual $\mathbf{r}_0$. The ratio of $\| \mathbf{r}_0 \| / \| \bar{\mathbf{r}}_0 \|$ is reported in columns two through six in table 4, as a function of enrichment vectors (0, 2, 4, 8, 16, 24) and Krylov vectors (30, 40, 50, 60, 70) between restarts. Clearly, preprocessing becomes more effective as more and higher quality information accumulates in the enrichment space.

6.1.4 Choosing the Ritz Vectors

Table 5 compares the efficacy of Ritz vector enrichment using two selection algorithms on the advection-diffusion problem. These results are obtained using the $ILU(2)$ preconditioner. Similar trends are recovered using $ILU(0)$ and $ILU(1)$, but are not reported herein. The format is the same as in tables 1, 2 and 3. At each restart, a harmonic Ritz procedure is used to determine the Ritz values, which are then assigned a merit base on their location in the complex plane. The k Ritz vectors that correspond to the values with the smallest merits are then chosen for enrichment. The two methods for assigning merit are $|\zeta_j| = [\sqrt{(\theta_r)_j^2 + (\theta_i)_j^2}]$ and $|\zeta_j| = [\frac{(\theta_r)_j}{\sqrt{(1-\theta_r)_j^2 + (\theta_i)_j^2}}]$. The two preconditioners that are used in the study are $ILU(0)$ and $ILU(2)$, while the diffusion coefficient is $\epsilon = \frac{1}{10}$.

Although sensitivity (up to a factor of 2) to the selection algorithms is noted, no clear winner emerges in this study. This emphasizes the need for a robust methodology to select Ritz vectors. Similar trends are observed using the conventional Ritz procedure to determine Ritz values.

6.2 Diffusion

6.2.1 Preliminaries

Results for the impact of enrichment on Laplace's equation are presented in tables 6, 7, 8, and 9. Tables 6, and 7 present results obtained on a 129^3 grid with Dirichlet BC's used on all six boundaries. Table 6 uses a uniform mesh, while table 7 uses a highly anisotropic mesh ($\epsilon_x = 10000, \epsilon_y = 100, \epsilon_z = 1$). The CPU time and iterations required to achieve the specified tolerance (10^{-10}), are presented as functions of 1) enrichment vectors (0, 2, 4) and 2) Krylov vectors (5, 10, 20, 40) between restarts. Two preconditioners are used in each study: $ILU(k), k = 0, 1$ and $Jac(k), k = 1, 2, 4, 8, 16, 32, 64$. The logarithm of the final residual replaces CPU-time for those cases that did not reach the specified tolerance in 2000 iterations. Galerkin projection initialization is not applicable because only a single $\mathbf{b}$ is solved in this test case.

6.2.2 Eigenvector Enrichment

The uniform grid study shown in table 6 demonstrates the benefits of enrichment for nearly all enrichment/preconditioners combinations. Enrichment greatly accelerates ($3\times$) the convergence for the weakly preconditioned cases [e.g., $Jac(1)$], with most of the benefit realized with two enrichment

vectors. The benefit of enrichment for strongly preconditioned cases [e.g., $Jac(64), ILU(1)$] is less significant ($2\times$).

As with preconditioner strength, the Krylov dimension strongly affects the efficacy of enrichment. The combination of 4 enrichment vectors with 5 Krylov vectors ([4 : 5]) proves to be ineffective, while the closely related cases [4 : 10] and [2 : 5] are accelerated relative to conventional GMRES. As the Krylov dimension approaches 40, the influence of enrichment is attenuated. Enrichment has no impact on the $ILU(1)$ / Krylov:40 combination.

An identical enrichment study performed on a highly anisotropic grid is presented in table 7. Although enrichment produces a small decrease in total iterations for most enrichment/preconditioner combinations, the actual CPU-time increases in many cases.

The uniform and anisotropic 129^3 grid trends shown in tables 6 and 7 are representative of other uniform and anisotropic grid studies (33^3, 65^3, and 257^3). Tables 8 and 9 compare 129^3 grid results with those obtained on grids of density 33^3, 65^3 and 257^3.

6.3 2D Unsteady Hexstream

6.3.1 Preliminaries

Tables 10, 11, 12, 13, 14, and 15 show comparisons of the impact of enrichment on an unsteady, 2D turbulent NS simulation. Table 10 shows the cumulative CPU time that is required for all 13 linear solutions (one full time step) to reach a residual of 10^{-10}, as a function of enrichment vectors, Krylov subspace vectors, preconditioners, and enrichment vector initialization. The first column in table 10 indicates the number of enrichment vectors (0, 2, 4, 6, 8, 10) that were used for the simulation; the first row immediately after the "preconditioner" label indicates the number of Krylov vectors (20, 25, 30, 35, 40) that were used between restarts. Columns $1-6$ and $7-11$ correspond to uninitialized and initialized Krylov space, respectively (see section 4.2.2, element IV). Numbers less than or equal to zero denote the logarithm of the final residual, which is given for simulations that fail to reach the specified tolerance after 2000 iterations. The three preconditioners used in the study are $ILU(2)$, $ILU(3)$, and $ILU(4)$. The preconditioners $ILU(0)$ and $ILU(1)$ could not reach the specified tolerance within a reasonable number of iterations. Harmonic Ritz vectors are used to enrich the subspace; the Ritz values were identified by using the k smallest $|\zeta_j|$ with the selection algorithm $|\zeta_j| = [\frac{\sqrt{((-0.25)-\theta_r)_j^2+(\theta_i)_j^2}}{\sqrt{(1-\theta_r)_j^2+(\theta_i)_j^2}}]$. (This selection algorithm was determined experimentally to identify the problematic Ritz values.)

6.3.2 Eigenvector Enrichment

Table 10 demonstrates the benefits of enrichment for the $ILU(2)$ and $ILU(3)$ preconditioners. Convergence is accelerated by approximately a factor of 2 for the $ILU(2)$ cases with less than 30 Krylov vectors. The impact of enrichment is less pronounced for the $ILU(4)$ preconditioner. Enrichment has the most impact on cases that were run with 30 or fewer Krylov vectors.

Table 11 shows the convergence study results for the $ILU(2)$ preconditioner. Only a single Ritz vector is enriched into the Krylov subspace at each restart. The eigenvalues that were used for enrichment are shown in the table, with the cumulative CPU time that was required to complete the 13 linear solutions. The eigenvalues with the greatest influence on the convergence are located near the left half-plane (LHP) point $(-0.25, 0.0)$. The impact decreases with increased distance from

the origin. Removing the eigenvalues inside the unit circle has even less impact on convergence. Finally, note that removal of the three eigenvalues that are furthermost from the unit circle (13.29, -23.28, 122.63) has little impact on convergence.

Figure 8 shows a plot of the *most* problematic eigenvector. It corresponds to the eigenvalue located at $(-0.16, 0, 0)$ in the complex plane. The eigenvector is related to an undamped pressure mode that is located near the blunt trailing edge of the airfoil.

6.3.3 Initialization

Initialization is accomplished by first enriching the space with Ritz vectors that are available from previous IRK iterative problems (section 4.2.2, element IV). Thus, the linear solution begins with the initial residual orthogonal to a preexisting enrichment space, which generally is not yet invariant. Second, the new problem is projected onto the existing solution space that is obtained from previous problems, which provides an optimal starting guess for the new problem (section 4.2.1). In table 10, columns $7-11$ show the cumulative CPU times that are required to solve all 13 linear problems. The initialization step is clearly not effective for this nonlinear unsteady NS simulation.

Table 12 shows the impact of projecting the current problem onto the existing solution space. Shown are the stage (IRK), the iteration within that stage and the reduction of the initial residual. Very little benefit is gained from this initial projection.

6.3.4 Choice of Ritz Vectors

Tables 13 and 14 compare different methods for choosing Ritz vectors, an important task that has received little attention in the enrichment literature. At restart, a p - dimensional Krylov space must be intelligently compressed, a task that is by no means trivial. Solution of the (standard or harmonic) Ritz problem yields p Ritz values that approximate the eigenvalues of the iteration matrix $\mathcal{A}$. Table 13 compares four different methods for assigning a value of merit to each Ritz value in the current iteration. The first column in table 13 indicates the number of enrichment vectors (0, 2, 4, 6, 8, 10) that are used for the simulation; columns $2-6$ and $7-11$ correspond to the number of Krylov vectors (20, 25, 30, 35, 40) that are used between restarts. All simulations are performed with the $ILU(2)$ preconditioner, and the Krylov subspace is not initialized based on past problems. Different merit functions $|\zeta_j|$ are used in each of the four cases.

The two merit functions $|\zeta_j|$ that perform well in this problem assign high merit to Ritz values near the origin, biased slightly into the LHP. These are $|\zeta_j| = [\frac{\sqrt{((-0.25)-\theta_r)_j^2+(\theta_i)_j^2}}{\sqrt{(1-\theta_r)_j^2+(\theta_i)_j^2}}]$ and $|\zeta_j| = [\frac{(\theta_r)_j}{\sqrt{(1-\theta_r)_j^2+(\theta_i)_j^2}}]$. The remaining two merit functions have difficulty locking onto the problematic modes. The function that performs the worst assigns merit based on inverse distance from the point $(1.0, 0.0)$.

The two common approaches for approximating matrix eigenvalues and eigenvectors are the harmonic Ritz and the conventional Ritz problems. See Appendix B for a discussion of the Ritz approximations. Table 14 compares these two approaches. The cumulative CPU time that is required to converge all 13 linear problems with an $ILU(2)$ preconditioner, is shown in the table. Columns $2-6$ and $7-11$ show the CPU times that are required with the use of the harmonic Ritz and conventional Ritz problem, respectively. The merit function that is used for both approaches

is $|\zeta_j| = [\frac{\sqrt{((-0.25)-\theta_r)_j^2+(\theta_i)_j^2}}{\sqrt{(1-\theta_r)_j^2+(\theta_i)_j^2}}]$. Both techniques perform well on this test case. Indeed, the CPU times that are obtained for each approach with the $ILU(3)$ and $ILU(4)$ preconditioners (not presented) are nearly indistinguishable.

This comparative study highlights the importance of the choice of merit function when approximating eigenvalues using either Ritz technique. Simply deflating eigenvalues near the origin may not be effective if the problematic eigenvalues are in the LHP or are far removed from the unit circle.

6.3.5 Reordering Strategies

Incomplete factorization preconditioners [e.g., $ILU(k)$] are known to be sensitive to matrix ordering [62–64]. The principle reason is that the preconditioner quality is sensitive to the size and number of terms that are dropped from the factorization, which depends indirectly on the ordering of the matrix. Experimental evidence presented thus far consistently demonstrates that enrichment depends to a large extent on the quality of the preconditioner. Table 15 presents a convergence study that compares the sensitivity of enrichment to matrix reordering, which directly tests this hypothesis.

The cumulative CPU time that is required for all linear solutions, is shown as a function of 1) enrichment vectors, 2) Krylov subspace vectors, 3) preconditioner, and 4) matrix ordering. Columns $2-6$ correspond to the boundary-layer reverse Cuthill-McKee (BL-RCM)[1] ordering that is used throughout the 2D unsteady Navier-Stokes study; columns $7-11$ correspond to standard RCM ordering. Note that the number of linear solutions that is required by the BL-RCM and RCM ordering is 13 and 11, respectively. Thus, a direct comparison favors the RCM method. Nevertheless, the significant reduction in CPU time that is required by the RCM ordering indicates that it is a significantly better ordering for the ILU(k) class of preconditioners $k > 2$. Furthermore, note that enrichment has little impact with the RCM ordered matrix, a result that is consistent with our hypothesis.

7 Conclusions

A general purpose algorithm, GMRES-E is presented to facilitate the reuse of Krylov subspace information to accelerate the convergence of linear systems with multiple right-hand sides $\mathcal{A}^i \mathbf{x} = \mathbf{b}^i$. The motivating problems for the algorithm are steady or unsteady simulations of nonlinear equations (e.g., unsteady NS). The kernel of the algorithm is a conventional GMRES algorithm but includes a specific form of eigenvector enrichment that facilitates arbitrary vectors. Four critical subelements contribute to the generality of the algorithm. First, solutions $\mathbf{x}^i$ from previous linear problems $\mathcal{A}\ \mathbf{x}^i = \mathbf{b}^i$ are stored, which allows for an optimal starting guess $\mathbf{x}_0$ via a Galerkin projection of the current problem onto the available solution space. Second, a variety of eigenvector approximation techniques are available. Third, each new problem requires compression or rotation of the subspace information that is available in an Arnoldi relation. A $Q\,R$ compression technique is shown to be effective for this purpose. Fourth, a Galerkin projection step is developed to inject

[1] Here, the grid boundary-layer cells are ordered separately from the Cartesian cells. (See also fig. 6).

the current residual $\mathbf{r}_0$ into the preprocessed Arnoldi relation. This step facilitates the admissibility of arbitrary enrichment vectors.

Two different problems are used to test the GMRES-E algorithm. The first is the steady, three-dimensional, advection-diffusion equation, which is simulated for various values of the diffusive parameter ϵ. The second is a two-dimensional, URANS (unsteady Reynolds-averaged NS) simulation of a wind turbine blade run at a high angle of attack.

General findings include the following: 1) Enrichment can significantly improve convergence and even eliminate stall in certain cases. 2) Eigenvector enrichment is ideally suited for cases in which a preconditioned matrix has several (e.g., less than 10) unclustered eigenvalues that the Krylov subspace cannot resolve. This scenario commonly occurs with "weak" preconditioners and small Krylov subspace dimensions. 3) Enrichment is ineffective or has no impact if many (i.e., 100) eigenvectors remain unclustered. Indeed, cases exist for which enrichment can even degrade the convergence. In short, enrichment is no substitute for a good preconditioner. 4) The Ritz value ranking and selection algorithm has a large impact on the convergence and the robustness of GMRES-E, as much if not moreso than the minimization technique (e.g., conventional/harmonic Ritz) that is used to approximate the eigenvalues. A general means for identifying problematic eigenvectors is not known, although selecting the Ritz values in the left half-plane near the origin is nearly always productive. 5) Reusing Ritz vectors when $\mathcal{A}$ changes (section 4.2.2) is not effective for the unsteady NS test case but is successful in the advection-diffusion tests, particularly for cases that are nearly symmetric (or definite). More work is needed to quantify the influence of matrix symmetry (and definiteness).

This work demonstrates the simplicity and flexibility of the proposed enrichment algorithm GMRES-E. Furthermore, it highlights the potential for subspace enrichment for cases that are derived from unsteady aerodynamics. Two important questions that still remain are: 1) "Depending on the application, which enrichment vectors are most productive?", and 2) "Do a priori or concurrent tests exist that can indicate when enrichment should *not* be used?"

A Which Portions of S_m and V_{m+1} are Krylov Subspaces?

We begin by deriving some preliminary results that are needed in the lemma to follow. First, left-multiplying $\mathcal{A}\ S_m \quad = \quad V_{m+1}\ \bar{\mathcal{H}}_m$ with $[\,V_{m+1}\,]^T$ and using the relationship $[\,V_{m+1}\,]^T\,V_{m+1} = \mathcal{I}_{m+1}$ immediately produces the relation

$$[\,V_{m+1}\,]^T\,\mathcal{A}\ S_m \qquad = \qquad \bar{\mathcal{H}}_m \quad . \tag{35}$$

In equation (17) we noted that S_m and V_{m+1} coincide for the integers in the interval $k \leq j \leq m$. Choosing j from this interval, using equation (35), and defining $\mathcal{P} = [\,\mathcal{I}_k \ - \ V_k\ {V_k}^T]$ produces the following relation

$$\begin{aligned} \mathcal{A}\,\mathbf{s}_j \ - \ \textstyle\sum_{i=1}^{k} \mathbf{v}_i\, h_{i,j} \quad &= \quad [\,\mathcal{I}_k \ - \ V_k\ {V_k}^T]\,\mathcal{A}\,\mathbf{s}_j \ + \ V_k\ {V_k}^T\,\mathcal{A}\,\mathbf{s}_j \ - \ \textstyle\sum_{i=1}^{k} \mathbf{v}_i\, h_{i,j} \\ &= \quad \mathcal{P}\,\mathcal{A}\,\mathbf{s}_j \ + \ \textstyle\sum_{i=1}^{k} \mathbf{v}_i([\mathbf{v}_i]^T\,\mathcal{A}\,\mathbf{s}_j \ - \ h_{i,j}) \\ &= \quad \mathcal{P}\,\mathcal{A}\,\mathbf{v}_j \end{aligned} \tag{36}$$

The exact character of the vector spaces S_m and V_{m+1} is quantified in the following lemma, which closely resembles the proof that is presented by Saad [56] for Arnoldi's method.

Lemma 5 *Assume that the algorithm that is defined in equation (10) uses k enrichment vectors and does not terminate before the mth step ($k < m$). Then, the vectors $\mathbf{v}_{k+1}, \mathbf{v}_{k+2}, \cdots, \mathbf{v}_m$ form an orthonormal basis for the Krylov subspace*

$$\mathcal{K}_{m-k} \;=\; span\{\mathbf{v}_{k+1}, (\mathcal{P}\,\mathcal{A}\,)\mathbf{v}_{k+1}, \cdots, (\mathcal{P}\,\mathcal{A}\,)^{m-k-1}\mathbf{v}_{k+1}\} \qquad .$$

Proof : The vectors $\mathbf{v}_j$ are orthonormal by construction. The fact that the last $m-k$ vectors span the space $\mathcal{K}_{m-k}$ follows from the fact that each vector $\mathbf{v}_j, \quad k+1 \le j \le m$, is of the form $q_{j-k-1}(\mathcal{P}\,\mathcal{A}\,)\mathbf{v}_{k+1}$, where q_{j-k-1} is a matrix polynomial of degree $j-k-1$. This can be shown by induction on j as follows. The result is clearly true for $j \;=\; k+1$, because $\mathbf{v}_{k+1} = q_0(\mathcal{P}\,\mathcal{A}\,)\mathbf{v}_{k+1}$ with $q_0(\mathcal{P}\,\mathcal{A}\,) \equiv \mathcal{I}$. Assume that the result is true for all integers $k+1, \cdots, j$ and consider $\mathbf{v}_{j+1}$. Using the definition of $h_{j+1,j}\mathbf{v}_{j+1}$ [see the Gram-Schmidt portion of the algorithm that is defined by equation (10)] with equation (36) leads to the relation

$$\begin{aligned} h_{j+1,j}\mathbf{v}_{j+1} &= \mathcal{A}\,\mathbf{s}_j - \textstyle\sum_{i=1}^{k} \mathbf{v}_i\, h_{i,j} - \sum_{i=k+1}^{j} \mathbf{v}_i\, h_{i,j} \\ &= \mathcal{P}\,\mathcal{A}\,\mathbf{v}_j - \textstyle\sum_{i=k+1}^{j} \mathbf{v}_i\, h_{i,j} \end{aligned} \tag{37}$$

The term $(\mathcal{P}\mathcal{A})q_{j-k-1}(\mathcal{P}\mathcal{A})\mathbf{v}_{j+1}$ is obviously a polynomial of degree $j-k$, while all $\mathbf{v}_i, k+1 \le i \le j$, are polynomials of degree $i-k-1$. Thus, $\mathbf{v}_{j+1}$ is a polynomial of degree $j-k$, which completes the proof. □

Remark. Note the two equivalent representations of $\mathcal{K}_m(S)$. The first is a Krylov space that is expanded in the vector $\mathbf{r}_\perp$ as

$$\mathcal{K}_m(S) \;=\; [\quad \xi_1, \quad \xi_2, \cdots, \quad \xi_k, \{\, \mathbf{r}_\perp\,, (\mathcal{P}\,\mathcal{A}\,)\,\mathbf{r}_\perp\,, (\mathcal{P}\,\mathcal{A}\,)^2\,\mathbf{r}_\perp\,, \cdots, (\mathcal{P}\,\mathcal{A}\,)^{m-k-1}\,\mathbf{r}_\perp\,\}]$$

The second is expanded in the vector $\mathbf{r}_0$ and then rotated as

$$\mathcal{K}_m(S) \;=\; [\quad \xi_1, \quad \xi_2, \cdots, \quad \xi_k, \mathcal{P}\{\, \mathbf{r}_0\,, (\,\mathcal{A}\,\mathcal{P})\,\mathbf{r}_0\,, (\,\mathcal{A}\,\mathcal{P})^2\,\mathbf{r}_0\,, \cdots, (\,\mathcal{A}\,\mathcal{P})^{m-k-1}\,\mathbf{r}_0\,\}]$$

by using the projection operator $\mathcal{P}$.

B Determination of Eigenvectors

Consider the non-symmetric eigenvalue problem

$$(\,\mathcal{A} \;-\; \theta\mathcal{I})\zeta \;=\; 0 \tag{38}$$

where $\mathcal{A}$ is the $n \times n$ matrix given in the linear problem $\mathcal{A}\;\mathbf{x} \;=\; \mathbf{b}\,.$

Projection techniques seek to extract an approximate solution to equation (38) from a vector subspace that in general may not include the exact eigenvector ζ. For example, if the subspace of candidate approximations (i.e., the search space) is S_m then an approximation to equation (38) is

$$[\,\mathcal{A} \;-\; \breve{\theta}\mathcal{I}]\,S_m\;\breve{\xi}_m \;=\; 0 \tag{39}$$

where $\breve{\xi}_m$ is an arbitrary m vector that combines the basis vectors in the search space S_m. In general, no vector $\breve{\xi}_m$ exists for any $\breve{\theta}$ that exactly satisfies equation (39). Note, however, that an exact eigenvector ζ_j in the space S_m exactly satisfies equation (39) for the value $\breve{\theta} \;=\; \theta$.

The harmonic Ritz problem is obtained by using a Petrov-Galerkin (or oblique projection) technique. Specifically, equation (39) is constrained such that any approximation error that exists is orthogonal to the subspace $\mathcal{A}\ S_m$. The resulting expression written in terms of one eigenpair $(\tilde{\theta},\ \tilde{\xi}_m)$ is

$$(\mathcal{A}\ S_m)^T\ [\mathcal{A}\ -\ \tilde{\theta}\mathcal{I}]\ S_m\ \tilde{\xi}_m\ =\ 0\quad . \tag{40}$$

Making use of the relationship $\mathcal{A}\ S_m\ =\ V_{m+1}\ \bar{\mathcal{H}}_m$ and simplifying yields

$$\begin{array}{lccccl} \{ & (\mathcal{A}\ S_m)^T\ \mathcal{A}\ S_m & - & \tilde{\theta} & (\mathcal{A}\ S_m)^T\ S_m & \}\ \tilde{\xi}_m\ = \\ \{ & (V_{m+1}\ \bar{\mathcal{H}}_m)^T\ V_{m+1}\ \bar{\mathcal{H}}_m & - & \tilde{\theta} & (V_{m+1}\ \bar{\mathcal{H}}_m)^T\ S_m & \}\ \tilde{\xi}_m\ = \\ \{ & (\bar{\mathcal{H}}_m)^T\ \bar{\mathcal{H}}_m & - & \tilde{\theta} & (\bar{\mathcal{H}}_m)^T\Gamma & \}\ \tilde{\xi}_m\ =\ 0 \end{array} \tag{41}$$

with $\Gamma\ =\ (V_{m+1})^T\ S_m$. The result $\{(\bar{\mathcal{H}}_m)^T\ \bar{\mathcal{H}}_m\ -\ \tilde{\theta}(\bar{\mathcal{H}}_m)^T\Gamma\}\ \tilde{\xi}_m\ =\ 0$ is the $m \times m$ generalized eigenvalue problem

$$\{\mathcal{A}_1\ -\ \tilde{\theta}\ \mathcal{A}_2\}\ \tilde{\xi}_m\ =\ 0\quad ;\quad \mathcal{A}_1\ =\ (\bar{\mathcal{H}}_m)^T\ \bar{\mathcal{H}}_m\quad ;\quad \mathcal{A}_2\ =\ (\bar{\mathcal{H}}_m)^T\Gamma$$

which is easily solved by using the Eispack [65] subroutine *rgg()*.

The close relationship between V_{m+1} and S_m make further simplification of Γ possible. Using the expressions for V_{m+1} and S_m found in equation (17) yields

$$\Gamma\ =\ \begin{bmatrix} (V_k)^T\ S_k & , & (V_k)^T\ V_{k+1\rightarrow m} \\ (V_{k+1\rightarrow m})^T\ S_k & , & (V_{k+1\rightarrow m})^T\ V_{k+1\rightarrow m} \\ (\mathbf{v}_{m+1})^T\ S_k & , & (\mathbf{v}_{m+1})^T\ V_{k+1\rightarrow m} \end{bmatrix} = \begin{bmatrix} (V_k)^T\ S_k & , & \mathbf{0} \\ (V_{k+1\rightarrow m})^T\ S_k & , & \mathcal{I}_{m-k} \\ (\mathbf{v}_{m+1})^T\ S_k & , & \mathbf{0} \end{bmatrix} \tag{42}$$

Thus, Γ is obtained at a cost of $k \times (m+1)$ additional dot products.

The condition number of the harmonic Ritz problem can be reduced by introducing the following algebraic simplifications. Recall that $\bar{\mathcal{H}}_m\ =\ \mathcal{H}_m\ +\ h_{m+1,m}\ \mathbf{e}_{m+1}\ \mathbf{e}_m{}^T$. Thus,

$$(\bar{\mathcal{H}}_m)^T\bar{\mathcal{H}}_m = (\mathcal{H}_m)^T\mathcal{H}_m+(h_{m+1,m})^2\mathbf{e}_m\mathbf{e}_m{}^T;\ (\bar{\mathcal{H}}_m)^T\Gamma = (\mathcal{H}_m)^T(V_m)^T S_m+h_{m+1,m}\mathbf{e}_m\mathbf{e}_{m+1}{}^T\Gamma$$

Multiplying equation (41) by the inverse transpose $(\mathcal{H}_m)^{-T}$ and defining $\mathbf{h_e}\ =\ (\mathcal{H}_m)^{-T}\mathbf{e}_m$ yields the equivalent harmonic Ritz eigenvalue formulation

$$\{\bar{\mathcal{A}}_1-\tilde{\theta}\ \bar{\mathcal{A}}_2\}\tilde{\xi}_m = 0\quad ;\quad \bar{\mathcal{A}}_1 = \mathcal{H}_m+(h_{m+1,m})^2\mathbf{h_e}\mathbf{e}_m{}^T\quad ;\quad \bar{\mathcal{A}}_2 = (V_m)^T S_m+(h_{m+1,m})\mathbf{h_e}\mathbf{e}_{m+1}{}^T\Gamma$$

.

B.1 Ritz-Value Error Estimation

Define the residual ρ_j in the eigenvalue problem (i.e., the eigenresidual) to be

$$\mathcal{A}\ \ S_m\ \ (\xi_m)_j\ -\ \theta\mathcal{I}\ S_m\ (\xi_m)_j\ =\ \rho_j. \tag{43}$$

We seek an efficient algorithm for determining $\|\rho_j\|_2$ for any approximate Ritz pairs $(\theta,\ (\xi_m))$ without additional matrix-vector operations (or a preconditioner step).

By definition, the L_2 norm of the complex eigenresidual ρ_j is defined as

$$(\xi_m)_j^{*T}[\mathcal{A}\ S_m\ -\ \theta^*\ S_m]^T\ [\mathcal{A}\ S_m\ -\ \theta\ S_m]\ (\xi_m)_j\ =\ \|\rho_j\|^2$$

and can be expanded into the form

$$(\xi_m)_j^{*T}[\,\mathcal{A}\;S_m\,]^T\;[\,\mathcal{A}\;S_m\;-\;\theta\,S_m\,]\,(\xi_m)_j\;\;-\;\;(\xi_m)_j^{*T}[\theta^*\,S_m\,]^T\;[\,\mathcal{A}\;S_m\;-\;\theta\,S_m\,]\,(\xi_m)_j\;=\;\|\rho_j\|^2 \tag{44}$$

where the superscript $*$ denotes a complex conjugate. The residual $\|\rho_j\|^2$ is composed of two terms. The first term is zero for the harmonic Ritz eigenpairs [eq. (40)] while the second is zero for conventional Ritz eigenpairs.

Now assume $(\,\mathcal{A}\;S_m\,)^T\,[\,\mathcal{A}\;-\;\theta\mathcal{I}]\,S_m\;(\xi_m)\;=\;\epsilon$ with ϵ close to machine precision. The Ritz pair $(\theta,\,(\xi_m)\,)$ will in general be complex valued and can be defined by $\theta\;=\;(\theta_r+\;i\;\theta_i)$ and $(\xi_m)\;=\;(\,(\xi_m)_r+\;i\;(\xi_m)_i)$. The L_2 norm of the eigenresidual ρ_j then reduces to

$$-\;(\xi_m)_j^{*T}[\theta^*\,S_m\,]^T\;[\,V_{m+1}\;\bar{\mathcal{H}}_m\;-\;\theta\,S_m\,]\,(\xi_m)_j\;=\;\|\rho_j\|^2 \tag{45}$$

Expanding the LHS of equation (45) yields

$$\begin{array}{lll}
- & [\theta_r\,(\xi_m)_r-\theta_i\,(\xi_m)_i]^T & \{[\,S_m{}^T V_{m+1}\;\bar{\mathcal{H}}_m\;-\;\theta_r\,S_m{}^T S_m\,]\,(\xi_m)_r\;+\;\theta_i\,S_m{}^T S_m\;(\xi_m)_i\} \\
- & [\theta_r\,(\xi_m)_i+\theta_i\,(\xi_m)_r]^T & \{[\,S_m{}^T V_{m+1}\;\bar{\mathcal{H}}_m\;-\;\theta_r\,S_m{}^T S_m\,]\,(\xi_m)_i\;-\;\theta_i\,S_m{}^T S_m\;(\xi_m)_r\} \\
+\;i & [\theta_r\,(\xi_m)_i+\theta_i\,(\xi_m)_r]^T & \{[\,S_m{}^T V_{m+1}\;\bar{\mathcal{H}}_m\;-\;\theta_r\,S_m{}^T S_m\,]\,(\xi_m)_r\;+\;\theta_i\,S_m{}^T S_m\;(\xi_m)_i\} \\
-\;i & [\theta_r\,(\xi_m)_r-\theta_i\,(\xi_m)_i]^T & \{[\,S_m{}^T V_{m+1}\;\bar{\mathcal{H}}_m\;-\;\theta_r\,S_m{}^T S_m\,]\,(\xi_m)_i\;-\;\theta_i\,S_m{}^T S_m\;(\xi_m)_r\}
\end{array} \tag{46}$$

which with further simplification, yields

$$\begin{array}{lllllll}
- & \theta_r & [& (\xi_m)_r{}^T S_m{}^T V_{m+1}\;\bar{\mathcal{H}}_m\;(\xi_m)_r & + & (\xi_m)_i{}^T S_m{}^T V_{m+1}\;\bar{\mathcal{H}}_m\;(\xi_m)_i &] \\
+ & \theta_i & [& (\xi_m)_i{}^T S_m{}^T V_{m+1}\;\bar{\mathcal{H}}_m\;(\xi_m)_r & - & (\xi_m)_r{}^T S_m{}^T V_{m+1}\;\bar{\mathcal{H}}_m\;(\xi_m)_i &] \\
+ & \|\theta\|^2 & [& (\xi_m)_r{}^T S_m{}^T S_m\;(\xi_m)_r & + & (\xi_m)_i{}^T S_m{}^T S_m\;(\xi_m)_i &] \;=\; \|\rho_j\|^2
\end{array} \tag{47}$$

Note that the (zero) imaginary component

$$\begin{array}{lllllll}
+ & i\,\theta_i[& (\xi_m)_r{}^T S_m{}^T V_{m+1}\;\bar{\mathcal{H}}_m\;(\xi_m)_r & + & (\xi_m)_i{}^T S_m{}^T V_{m+1}\;\bar{\mathcal{H}}_m\;(\xi_m)_i &] \\
+ & i\,\theta_r[& (\xi_m)_i{}^T S_m{}^T V_{m+1}\;\bar{\mathcal{H}}_m\;(\xi_m)_r & - & (\xi_m)_r{}^T S_m{}^T V_{m+1}\;\bar{\mathcal{H}}_m\;(\xi_m)_i &] \;=\; 0
\end{array} \tag{48}$$

can be used as a measure of roundoff error.

B.2 Ritz-Value Selection Algorithms

The harmonic Ritz minimization procedure is generally considered to accurately predict the small eigenvalues in the matrix [19, 26]. The optimal choice for a selection algorithm would seem to be the selection of the k smallest (in magnitude) harmonic Ritz values. Frequently, this is the case, but exceptions do exist.

In most applications, the matrix $\mathcal{A}$ is a matrix product of a discretization and a preconditioner matrix (i.e., $\mathcal{A}=\;\bar{\mathcal{A}}\,M^{-1}$). An effective preconditioner clusters the eigenvalues of $\bar{\mathcal{A}}$ within the unit circle that is centered in the complex plane at the point $(1.0, 0.0)$. Occasionally, however, eigenvalues are not confined to within this unit circle. If they are near the origin, then the standard selection algorithm identifies them as problematic. Another selection algorithm is necessary, however, if they are far removed from the origin.

The selection algorithm begins with a weighting step that assigns each the Ritz value θ_j a merit ζ_j. The discrete function ζ_j is constructed such that it preferentially assigns small merit to Ritz

values depending on their location in the complex plane. Next, the Ritz values are sorted according to their merit, and the k smallest values are chosen as enrichment vectors.

Four different routines are used in the GMRES-E algorithm. Unfortunately, no merit function has been found to be optimal for all problems. What is clear, however, is that the choice of merit function has a significant influence on the overall performance of GMRES-E. The merit routines are

$$\begin{aligned} |\zeta_j| &= [\sqrt{(\theta_r)_j^2 + (\theta_i)_j^2}] \\ |\zeta_j| &= [\frac{1}{\sqrt{(1-\theta_r)_j^2 + (\theta_i)_j^2}}] \\ |\zeta_j| &= [\frac{-(\theta_r)_j}{\sqrt{(1-\theta_r)_j^2 + (\theta_i)_j^2}}] \\ |\zeta_j| &= [\frac{\sqrt{((-0.25)-\theta_r)_j^2 + (\theta_i)_j^2}}{\sqrt{(1-\theta_r)_j^2 + (\theta_i)_j^2}}] \end{aligned} \tag{49}$$

The first algorithm (i.e., the conventional method of selection) selects Ritz values based on their distance from the origin $(0.0, 0.0)$. The second algorithm selects values based on their inverse distance from the point $(1.0, 0.0)$. This technique overemphasizes eigenvalues that are outside the unit circle. The third algorithm selects the Ritz values base on their inverse distance from the position $(1.0, 0.0)$ by assigning higher importance to those in the left half-plane. The fourth algorithm emphasizes Ritz values outside the unit circle but focuses on those located near the point $(-0.25, 0.0)$.

The selection algorithm must be modified slightly when a complex conjugate eigenpair is assigned the ranks of k and $k+1$. Including k vectors would require inclusion of only one of the pair. In these cases, the number of retained eigenvalues is reduced to $k-1$.

Now, assemble an $n \times k$ matrix $\tilde{\mathcal{P}}_k$ with the columns composed of linear combinations of the harmonic Ritz eigenvectors [see eq. (41)]. For the complex conjugate eigenvector pairs, use the (linearly dependent) Schur vectors rather than the eigenvectors, a modification that allows the use of real arithmetic while retaining the same column span for $\tilde{\mathcal{P}}_k$. The Schur vectors are formed from the conjugate eigenvector pairs $\tilde{\xi}_{n\ j}$, $\tilde{\xi}_{n\ j+1}$ using the rotations

$$\begin{bmatrix} \acute{\xi}_{n\ j} \\ \acute{\xi}_{n\ j+1} \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 1 & 1 \\ i & -i \end{bmatrix} \begin{bmatrix} \tilde{\xi}_{n\ j} \\ \tilde{\xi}_{n\ j+1} \end{bmatrix} \quad . \tag{50}$$

With $\tilde{\mathcal{P}}_k$ assembled, use the modified Gram-Schmidt algorithm on $\tilde{\mathcal{P}}_k$ to produce the orthogonal matrix $\bar{\mathcal{P}}_k$. The columns of the matrix $\bar{\mathcal{P}}_k$ are no longer eigenvectors (or Schur vectors) of the (harmonic) Ritz problem, but the column span $\bar{\mathcal{P}}_k$ is the same as for the original eigenvectors.

References

1. Eisenstat, S. C.; Elman, H. C.; and Shultz, M. H.: Variational iterative methods for nonsymmetric systems of linear equations. *SIAM J. Numer. Anal.*, vol. 20, no. 2, 1983, pp. 345–357.

2. Saad, Y.; and Schultz, M. H.: GMRES: A generalized minimum residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. Statist. Comput.*, vol. 7, 1986, pp. 856–869.

3. van der Vorst, H.: BI-CGSTAB: A fast and smoothly converging variant of BI-CG for the solution of nonsymmetric linear systems. *SIAM J. Sci. Stat. Comput.*, vol. 13, no. 2, 1992, pp. 631–644.

4. Saad, Y.: A flexible inner-outer preconditioned GMRES algorithm. *SIAM J. Sci. Statist. Comput.*, vol. 14, 1993, pp. 461–469.

5. van der Vorst, H.; and Vuik, C.: GMRESR: a family of nested GMRES methods. *Numer. Lin. Alge. Appl.*, vol. 1, no. 4, 1994, pp. 369–386.

6. Brown, P. N.; and Saad, Y.: Hybrid Krylov methods for nonlinear systems of equations. *SIAM J. Sci. Stat. Comput.*, vol. 11, 1990, pp. 450–481.

7. Gropp, W. D.; and Keyes, D. E.: Domain decomposition methods in computational fluid dynamics. *I.J. Num. Meth. Fluids*, vol. 14, 1992, pp. 147–165.

8. Blanco, M.; and Zingg, D. W.: Fast Newton-Krylov method for unstructured grids. *AIAA J.*, vol. 36, no. 4, 1998, pp. 607–612.

9. Vuik, C.; van Nooyen, R. R. P.; and Wesseling, P.: Practical aspects of parallelism in ILU-preconditioned GMRES. *Parallel Computing*, vol. 24, 1998, pp. 1927–1946.

10. PETSc-FUN3D home page. http://www-fp.mcs.anl.gov/petsc-fun3d/, 2009.

11. Chapman, A.; Saad, Y.; and L., W.: High-order ILU preconditioners for CFD problems. *Int. J. Numer. Meth. Fluids*, vol. 33, 2000, pp. 767–788.

12. Cai, X.-C.; and Keyes, D. E.: Nonlinearly Preconditioned Inexact Newton Algorithms. *SIAM J. Sci. Comput.*, vol. 24, no. 1, 2002, pp. 183–200.

13. Knoll, D. A.; and Keyes, D. E.: Jacobian-free Newton-Krylov methods: a survey of approaches and applications. *J. Comput. Phys.*, vol. 193, 2004, pp. 357–397.

14. Patel, A.: Développement d'un solveur adaptif sur maillages non-sturcturés hexaédriques. Ph.D. Thesis, Université Libre de Bruxelles, 2003.

15. Patel, A.; Léonard, B.; Elsden, M.; and Hirsch, C.: A parallel multigrid adaptive industrial flow solver on all-hexahedra unstructured meshes. *International conference on adaptive modeling and simulation (ADMOS), Barcelona, Spain*, 2003.

16. Shroff, G. M.; and Keller, H. B.: Stabilization of unstable procedures: The recursive projection method. *SIAM J. Numer. Anal.*, vol. 30, no. 4, 1993, pp. 1099–1120.

17. Kharchenko, S. A.; and Yeremin, A. Y.: Eigenvalue translation based preconditioners for the GMRES(k) method. *Numer. Linear Algebra Appl.*, vol. 2, 1995, pp. 51–77.

18. Erhel, J.; Burrage, K.; and Pohl, B.: Restarted GMRES preconditioned by deflation. *J. Comput. Appl. Math.*, vol. 69, 1996, pp. 303–318.

19. Chapman, A.; and Saad, Y.: Deflated and augmented Krylov subspace techniques. *Numer. Linear Algebra Appl.*, vol. 4, 1997, pp. 43–66.

20. Burrage, K.; Erhel, J.; Pohl, B.; and Williams, A.: A deflation technique for linear systems of equations. *SIAM J. Sci. Comput.*, vol. 19, 1998, pp. 1245–1260.

21. Burrage, K.; and Erhel: On the Performance of Various Adaptive Preconditioned GMRES Strategies. *Numer. Linear Algebra Appl.*, vol. 5, 1998, pp. 101–121.

22. Baglama, J.; Calvetti, D.; Golub, G. H.; and Reichel, L.: Adaptively preconditioned GMRES algorithms. *SIAM J. Sci. Comput.*, vol. 20, 1999, pp. 243–269.

23. Sorensen, D. C.: Implicit application of polynomial filters in a k-step Arnoldi method. *SIAM J. Matrix Anal. Appl.*, vol. 13, 1992, pp. 357–385.

24. Morgan, R. B.: A restarted GMRES method augmented with eigenvectors. *SIAM J. Matrix Anal. Appl.*, vol. 16, 1995, pp. 1154–1171.

25. Morgan, R. B.: Implicitly restarted GMRES and Arnoldi methods for nonsymmetric systems of equations. *SIAM J. Matrix Anal. Appl.*, vol. 21, no. 4, 2000, pp. 1112–1135.

26. Morgan, R. B.: GMRES with deflated restarting. *SIAM J. Sci. Comput*, vol. 24, no. 1, 2002, pp. 20–37.

27. Morgan, R. B.; and Zeng, M.: Harmonic projection methods for large non-symmetric eigenvalue problems. *Numer. Linear Algebra Appl.*, vol. 5, 1998, pp. 33–55.

28. Saad, Y.: Analysis of augmented Krylov subspace methods. *SIAM J. Matrix Anal. Appl.*, vol. 18, 1997, pp. 435–449.

29. van der Vorst, H.: An iterative solution method for solving f(A)x=b, using Krylov subspace information obtained for the symmetric positive definite matrix A. *J. Comput. Appl. Math.*, vol. 18, 1987, pp. 249–263.

30. Smith, C. F.; Peterson, A. F.; and Mittra, R.: A conjugate gradient algorithm for the treatment of multiple incident electromagnetic Fields. *IEEE Trans. Antennas and Propagation*, vol. 37, 1989, pp. 1490–1493.

31. Papadrakakis, M.; and Smerou, S.: A new implementation of the Lanczos method in linear problems. *I. J. Numer. Methods Engrg.*, vol. 29, 1990, pp. 141–159.

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

33. Chan, T. F.; and Wan, W. L.: Analysis of projection methods for solving linear systems with multiple right-hand sides. *SIAM J. Sci. Comput.*, vol. 1, no. 6, 1997, pp. 1698–1721.

34. Morgan, R. B.: Restarted block-GMRES with deflation of eigenvalues. *Appl. Numer. Math.*, vol. 54, 2005, pp. 222–236.

35. Erhel, J.; and Guyomarc'h, F.: An augmented conjugate gradient method for solving consecutive symmetric positive definite linear systems. *SIAM J. Matrix Anal.*, vol. 21, 2000, pp. 1279–1299.

36. Golub, G.; Ruiz, D.; and Touhami, A.: A hybrid approach combining Chebyshev filter and conjugate gradient for solving linear systems with multiple right-hand sides. *SIAM J. Matrix Anal. Appl.*, vol. 29, no. 3, 2006, pp. 774–795.

37. Parks, M. L.; De Sturler, E.; Mackey, G.; Johnson, D. D.; and Maiti, S.: Recycling Krylov subspaces for sequences of linear systems. *SIAM J. Sci. Comput.*, vol. 28, 2006, pp. 1651–1674.

38. Rey, C.; and Risler, F.: A Rayleigh-Ritz preconditioner for the iterative solution to large scale nonlinear problems. *Numer. Algorithms*, vol. 17, 1998, pp. 279–311.

39. Risler, F.; and Rey, C.: Iterative accelerating algorithms with Krylov subspaces for the solution to large-scale nonlinear problems. *Numer. Algorithms*, vol. 23, 2000, pp. 1–30.

40. Tromeur-Dervout, D.; and Vassilevsi, Y.: Choice of initial guess in iterative solution of series of systems arising in fluid flow simulations. *J. Comp. Phys.*, vol. 219, 2006, pp. 210–227.

41. Fischer, P.: Projection techniques for iterative solution of A x = b with successive right-hand sides. *Comput. Meth. Appl. Meth. Engrg.*, vol. 163, 1998, pp. 193–204.

42. De Sturler, E.: Truncation strategies for optimal Krylov subspace methods. *SIAM J. Numer. Anal.*, vol. 36, 1999, pp. 864–889.

43. Wu, K.; and Simon, H.: Thick-restart Lanczos method for large symmetric eigenvalue problems. *SIAM J. Matrix Anal. Appl.*, vol. 22, 2000, pp. 602–616.

44. Parlett, B. N.: *The symmetric Eigenvalue Problem*. Prentice-Hall, Englewood Cliffs, N.J., 1980.

45. Nicolaides, R. A.: Deflation of conjugate gradients with applications to boundary value problems. *SIAM J. Numer. Anal.*, vol. 24, 1987, pp. 355–365.

46. Mansfield, L.: Damped Jacobi preconditioning and coarse grid deflation for conjugate gradient iteration on parallel computers. *SIAM J. Sci. Statist. Comput.*, vol. 12, 1991, pp. 1314–1323.

47. Vuik, C.; Segal, A.; and Meijerink, J. A.: An efficient preconditioned CG method for the solution of a class of layered problems with extreme contrasts in the coefficients. *J. Comp. Phys.*, vol. 152, 1999, pp. 385–403.

48. Vuik, C.; Segal, A.; Meijerink, J. A.; and Wijma, G. T.: The construction of projection vectors for a deflated ICCG method applied to problems with extreme contrasts in the coefficients. *J. Comp. Phys.*, vol. 172, 2001, pp. 426–450.

49. Vuik, C.; and Frank, J.: Coarse grid acceleration of a parallel block preconditioner. *Future Generation Computer Systems*, vol. 17, 2001, pp. 933–940.

50. Frank, J.; and Vuik, C.: On the construction of deflation-based preconditioners. *SIAM J. Sci. Comp.*, vol. 23, 2001, pp. 442–462.

51. Lottes, J. W.; and Fischer, P. F.: Hybrid Multigrid/Schwarz Algorithms for the Spectral Element Method. *J. Sci. Comput.*, vol. 24, no. 1, 2005, pp. 45–78.

52. Verkaik, J.; Vuik, C.; Paarhuis, B.; and Twerda, A.: The deflation accelerated Schwarz method for CFD. *Lecture Notes in Computer Science*, vol. 3514, 2005, pp. 868–875.

53. Nabben, R.; and Vuik, C.: A comparison of deflation and the balancing preconditioner. *SIAM J. Sci. Comput.*, vol. 27, no. 5, 2006, pp. 1742–1759.

54. Tang, J. M.; and Vuik, C.: An efficient deflation method applied on 2-D and 3-D bubbly flow problems. *Electronic Transactions on Numerical Analysis*, vol. 26, no. 06-01, 2007, pp. 330–349.

55. Vuik, C.; Segal, A.; el Yaakoubi, L.; and Dufour, E.: A comparison of various deflation vectors applied to elliptic problems with discontinuous coefficients. *Appl. Num. Math.*, vol. 41, 2002, pp. 219–233.

56. Saad, Y.: *Iterative methods for sparse linear systems*. Copyright: Yousef Saad, 2000.

57. Kilmer, M. E.; and De Sturler, E.: Recycling subspace information for diffuse optical tomography. *SIAM J. Sci. Comp.*, vol. 27, 2006, pp. 2140–2166.

58. Saad, Y.: SPARSKIT: a basic tool kit for sparse matrix computations, Version 2. *http://www-users.cs.umn.edu/s̃aad/software/SPARSKIT/sparskit.html*, 1994.

59. Spalart, P. R.; and Allmaras, S. R.: A One-Equation Turbulence Model for Aerodynamic Flows. *Techerche Aerospatiale*, vol. 1, 1994, pp. 5–214.

60. Bijl, H.; Carpenter, M.; Vatsa, V.; and Kennedy, C.: Implicit Time Integration Schemes for the Unsteady Compressible Navier-Stokes Equations: Laminar Flow. *J. Comput. Phys.*, vol. 179, 2002, pp. 313–329.

61. Carpenter, M.; Kennedy, C.; Bijl, H.; Viken, S.; and Vatsa, V.: Fourth-order Runge-Kutta schemes for fluid mechanics applications. *J. Sci. Comput.*, vol. 25, 2005, pp. 157–194.

62. Dutto, L. C.: The effect of ordering on preconditioned GMRES algorithm, for solving the compressible Navier-Stokes equations. *Int. J. Numer. Methods Eng.*, vol. 36, 1993, p. 457.

63. Benzi, M.; Joubert, W. D.; and Mateescu, G.: Numerical experiments with parallel orderings for ILU preconditioners. *Electron. Trans. Numer. Anal.*, vol. 8, 1999, p. 88.

64. Benzi, M.: Preconditioning Techniques for Large Linear Systems: A Survey. *J. Comput. Phys.*, vol. 182, 2002, pp. 418–477.

65. *www.netlib.org/eispack*.

Enrichment Vectors	For every new problem: discard $\mathcal{A}\ S_m{}^{j-1} = V_{m+1}{}^{j-1}\,\bar{\mathcal{H}}_m{}^{j-1}$					For every new problem, pre-enrich: retain $\mathcal{A}\ S_m{}^{j-1} = V_{m+1}{}^{j-1}\,\bar{\mathcal{H}}_m{}^{j-1}$				
	Preconditioner: ILU(0)									
	Maximum Krylov subspace dimension									
	30	40	50	60	70	30	40	50	60	70
0	0	0	-1	-1	-1	0	0	0	-1	-1
2	-3	-2	-3	-5	-5	-4	-4	-3	-5	-4
4	-4	-7	-5	-8	-8	-4	-4	-5	-7	-8
8	-8	-9	650	681	601	-7	176	524	507	395
16	555	530	460	447	421	106	446	335	293	261
24	-3	671	570	523	485	625	231	177	162	153
	Preconditioner: ILU(2)									
	Maximum Krylov subspace dimension									
	30	40	50	60	70	30	40	50	60	70
0	-2	-2	-3	-3	-3	-2	-2	-3	-3	-3
2	-5	-8	-9	1052	953	-5	-8	279	945	910
4	748	822	518	628	492	496	726	481	579	436
8	461	363	368	381	357	377	246	262	226	226
16	432	255	279	255	249	341	183	154	143	128
24	686	358	309	283	306	275	132	98	144	91

Table 1. Comparison of effectiveness of enrichment for advection-diffusion test problem with $\epsilon = 1$. The CPU time required to achieve the specified tolerance (1.0^{-10}), is presented as a function of enrichment vectors (0, 2, 4, 8, 16, 24) and Krylov vectors (30, 40, 50, 60, 70) between restarts. Negative numbers indicate stalled iterations, whereby the logarithm of the residual at iteration 2000 is given. Additional parameters include preconditioners [$ILU(0)$, $ILU(1)$, and $ILU(2)$] and enrichment vector initialization.

Enrichment Vectors	For every new problem: discard $\mathcal{A}\ S_m{}^{j-1} = V_{m+1}{}^{j-1}\,\bar{\mathcal{H}}_m{}^{j-1}$					For every new problem, pre-enrich: retain $\mathcal{A}\ S_m{}^{j-1} = V_{m+1}{}^{j-1}\,\bar{\mathcal{H}}_m{}^{j-1}$				
	Preconditioner: $ILU(0)$									
	Maximum Krylov subspace dimension									
	30	40	50	60	70	30	40	50	60	70
0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	-3	0	0	0	0	-3
8	0	493	651	590	540	0	405	498	455	401
16	375	343	345	322	369	199	160	162	159	160
24	-5	544	486	440	431	527	222	177	179	164
	Preconditioner: $ILU(2)$									
	Maximum Krylov subspace dimension									
	30	40	50	60	70	30	40	50	60	70
0	0	0	0	0	0	0	0	0	0	-3
2	0	0	0	0	746	0	0	0	0	583
4	0	-6	683	643	573	213	-2	586	482	439
8	457	348	321	267	237	323	257	256	242	215
16	233	198	196	217	210	110	95	98	99	97
24	487	306	264	254	259	236	118	104	106	93

Table 2. Comparison of effectiveness of enrichment for advection-diffusion test problem with $\epsilon = \frac{1}{10}$. (See table 1 for further details).

Enrichment Vectors	For every new problem: discard $\mathcal{A}\ S_m{}^{j-1} = V_{m+1}{}^{j-1}\,\bar{\mathcal{H}}_m{}^{j-1}$					For every new problem, pre-enrich: retain $\mathcal{A}\ S_m{}^{j-1} = V_{m+1}{}^{j-1}\,\bar{\mathcal{H}}_m{}^{j-1}$				
	Preconditioner: $ILU(0)$									
	Maximum Krylov subspace dimension									
	30	40	50	60	70	30	40	50	60	70
0	0	0	540	592	607	0	0	483	499	492
2	259	268	325	487	398	203	225	253	305	281
4	313	305	293	362	397	215	321	262	314	330
8	212	244	256	314	311	207	196	232	227	241
16	293	267	253	256	257	252	243	247	243	250
24	595	385	325	303	288	549	354	310	291	293
	Preconditioner: $ILU(2)$									
	Maximum Krylov subspace dimension									
	30	40	50	60	70	30	40	50	60	70
0	337	307	285	126	123	314	253	206	152	123
2	190	178	186	103	103	148	160	133	101	99
4	160	209	142	105	104	173	153	117	96	91
8	149	147	129	97	98	142	136	112	102	100
16	165	133	123	107	98	169	132	133	123	118
24	294	166	136	117	108	290	180	165	150	143

Table 3. Comparison of effectiveness of enrichment for advection-diffusion test problem with $\epsilon = \frac{1}{100}$. (See table 1 for further details).

Enrichment	Maximum Krylov subspace dimension				
vectors	30	40	50	60	70
0	0.96	0.81	0.76	0.60	0.56
2	0.68	0.57	0.51	0.54	0.56
4	0.49	0.53	0.46	0.45	0.44
8	0.30	0.56	0.54	0.44	0.32
16	0.24	0.22	0.23	0.23	0.21
24	0.13	0.15	0.15	0.15	0.14

Table 4. Comparison of effectiveness of projection initialization on advection-diffusion test problem with $\epsilon = \frac{1}{10}$. The starting residual (normalized to 1) immediately after the projection, is presented as a function of enrichment vectors (0, 2, 4, 8, 16, 24) and Krylov vectors (30, 40, 50, 60, 70) between restarts.

Enrichment vectors	For every new problem: discard $\mathcal{A}\ S_m{}^{j-1} = V_{m+1}{}^{j-1}\,\bar{\mathcal{H}}_m{}^{j-1}$					For every new problem, pre-enrich: retain $\mathcal{A}\ S_m{}^{j-1} = V_{m+1}{}^{j-1}\,\bar{\mathcal{H}}_m{}^{j-1}$				
	$\lvert\zeta_j\rvert = [\sqrt{(\theta_r)_j^2 + (\theta_i)_j^2}]$									
	Maximum Krylov subspace dimension									
	30	40	50	60	70	30	40	50	60	70
0	0	0	0	0	0	0	0	0	0	-3
2	0	0	0	0	746	0	0	0	0	583
4	0	-6	683	643	573	213	-2	586	482	439
8	457	348	321	267	237	323	257	256	242	215
16	233	198	196	217	210	110	95	98	99	97
24	487	306	264	254	259	236	118	104	106	93
	$\lvert\zeta_j\rvert = [\frac{(\theta_r)_j}{\sqrt{(1-\theta_r)_j^2 + (\theta_i)_j^2}}]$									
	Maximum Krylov subspace dimension									
	30	40	50	60	70	30	40	50	60	70
0	0.	0.	0.	0.	-1.	0.	0.	0.	0.	-2.
2	0.	0.	-1.	-1.	-2.	0.	0.	-1.	-2.	-5.
4	0.	-1.	332.	611.	603.	0.	0.	554.	585.	600.
8	246.	238.	249.	245.	263.	169.	165.	176.	183.	188.
16	254.	225.	215.	225.	244.	120.	116.	116.	116.	132.
24	556.	331.	290.	275.	276.	261.	134.	119.	121.	126.

Table 5. Comparison of effectiveness of enrichment for advection-diffusion test problem with $\epsilon = \frac{1}{10}$. Eigenvalues are deflated starting with negative-most (LHS) eigenvalues. The preconditioner is $ILU(2)$. The CPU time required to reach the specified tolerance is tabulated.

Enrichment	Uniform Grid: $129 \times 129 \times 129$											
Vectors	Enrichment vectors : 0				Enrichment vectors : 2				Enrichment vectors : 4			
	CPU-Time											
Sweeps	5	10	20	40	5	10	20	40	5	10	20	40
0	-1.	-3.	-3.	-9.	-3.	1158.	743.	749.	-1.	1222.	630.	638.
1	-5.	1298.	938.	910.	1069.	400.	301.	351.	-2.	415.	274.	304.
2	-4.	-8.	1416.	1234.	-9.	699.	426.	438.	-2.	617.	368.	400.
4	-7.	1778.	1167.	925.	1443.	524.	380.	409.	-3.	509.	319.	343.
8	3030.	1575.	988.	579.	1262.	488.	358.	351.	-6.	444.	301.	316.
16	2815.	1501.	847.	496.	1169.	500.	386.	363.	3523.	450.	342.	331.
32	2755.	1389.	693.	393.	1129.	533.	418.	368.	3169.	481.	371.	362.
64	3018.	1124.	622.	409.	1191.	617.	500.	408.	2980.	556.	500.	409.
$ILU(0)$	755.	476.	388.	324.	554.	233.	168.	218.	-5.	250.	185.	231.
$ILU(1)$	436.	282.	233.	172.	327.	281.	132.	175.	-8.	182.	143.	179.
	Number of iterations											
Sweeps	5	10	20	40	5	10	20	40	5	10	20	40
0	2000	2000	2000	2000	2000	1950	920	600	2000	1940	720	480
1	2000	1960	1080	720	1690	560	320	280	2000	570	280	240
2	2000	2000	1380	840	2000	710	400	280	2000	720	340	280
4	2000	1560	880	520	1350	460	280	240	2000	470	240	200
8	1675	890	520	280	765	280	200	160	2000	290	160	160
16	885	500	280	160	415	180	140	120	1920	180	120	120
32	465	250	140	80	225	110	80	80	985	110	80	80
64	245	110	80	40	120	70	60	40	500	70	60	40
$ILU(0)$	1545	840	500	280	1020	370	200	200	2000	380	220	200
$ILU(1)$	715	410	260	160	500	410	140	160	2000	250	160	160

Table 6. Comparison of the effectiveness of enrichment on the LaPlace's equation. The diffusion coefficients used in the study are $\epsilon_x = \epsilon_y = \epsilon_z = 1$. The grid is $129 \times 129 \times 129$ and Dirichlet boundary conditions are used on all six boundaries. The CPU time required and iterations required to achieve the specified tolerance (1.0^{-10}), are presented functions of 1) enrichment vectors (0,2,4) and 2) Krylov vectors (5,10,20,40) between restarts, and preconditioners. The Jacobi iterations are used as a precondition with varying number of subiterations (0,1,2,4,8,16,32,64). Negative times indicate stalled iterations, whereby the logarithm of the residual at iteration 2000 is given.

Enrichment	10000:100:1 Aspect Grid: $129 \times 129 \times 129$											
Vectors	Enrichment vectors : 0				Enrichment vectors : 2				Enrichment vectors : 4			
	CPU-Time											
Sweeps	5	10	20	40	5	10	20	40	5	10	20	40
0	-1.	-3.	-5.	-10.	-2.	-4.	-7.	2157.	-1.	-4.	-8.	2114.
1	-5.	1308.	900.	851.	-4.	1451.	883.	856.	-2.	-9.	955.	873.
2	-4.	-8.	1390.	1175.	-4.	-8.	1309.	1163.	-2.	-7.	1309.	1166.
4	-7.	1797.	1077.	977.	-5.	1867.	1071.	961.	-2.	-10.	1111.	970.
8	3061.	1568.	941.	870.	-7.	1604.	963.	874.	-3.	1917.	968.	876.
16	2916.	1351.	951.	881.	4558.	1532.	966.	877.	-3.	1711.	967.	892.
32	2763.	1470.	1057.	994.	4227.	1597.	1063.	980.	-6.	1764.	1070.	985.
64	2606.	1650.	1302.	1180.	4511.	1797.	1324.	1201.	-8.	1953.	1310.	1217.
$ILU(0)$	63.	57.	68.	100.	138.	75.	76.	105.	794.	102.	89.	112.
$ILU(1)$	66.	59.	68.	98.	133.	75.	77.	99.	717.	99.	85.	105.
	Number of iterations											
Sweeps	5	10	20	40	5	10	20	40	5	10	20	40
0	2000	2000	2000	2000	2000	2000	2000	1720	2000	2000	2000	1600
1	2000	1960	1040	640	2000	1980	920	640	2000	2000	940	600
2	2000	2000	1340	800	2000	2000	1180	760	2000	2000	1180	720
4	2000	1570	820	560	2000	1590	780	520	2000	2000	800	520
8	1705	890	500	400	2000	920	500	360	2000	1240	520	360
16	915	450	320	280	1625	530	320	280	2000	690	340	280
32	465	270	200	200	830	310	200	200	2000	410	220	200
64	230	160	140	120	445	190	140	120	2000	240	140	120
$ILU(0)$	135	100	100	120	250	120	100	120	1605	160	100	120
$ILU(1)$	110	90	80	80	205	110	80	80	1280	140	100	80

Table 7. Comparison of the effectiveness of enrichment on the diffusion test problem. The diffusion coefficients used in the study are $\epsilon_x = 10000$; $\epsilon_y = 100$; $\epsilon_z = 1$. All test parameters are identical to those used in the uniform grid case (see table 6).

Enrichment	1:1:1 Aspect Grid: $ILU(0)$											
Vectors	Enrichment vectors : 0				Enrichment vectors : 2				Enrichment vectors : 4			
	CPU-Time											
Grid	5	10	20	40	5	10	20	40	5	10	20	40
033^3	< 1	< 1	1.	1.	< 1	< 1	1.	1.	1.	< 1	1.	1.
065^3	23.	13.	14.	10.	45.	22.	8.	21.	204.	12.	9.	11.
129^3	755.	476.	388.	324.	554.	233.	168.	218.	-5.	250.	185.	231.
257^3	-3.	-6.	11575.	NA	-7.	6254.	4339.	NA	-2.	6658.	3658.	NA
	Number of iterations											
	5	10	20	40	5	10	20	40	5	10	20	40
033^3	15	10	20	40	15	10	20	40	35	10	20	40
065^3	200	140	120	120	355	150	120	120	1790	200	120	120
129^3	1545	840	500	280	1020	370	200	200	2000	380	220	200
257^3	2000	2000	1680	NA	2000	1110	580	NA	2000	1140	460	NA
Enrichment	1:1:1 Aspect Grid: $ILU(1)$											
Vectors	Enrichment vectors : 0				Enrichment vectors : 2				Enrichment vectors : 4			
	CPU-Time											
	5	10	20	40	5	10	20	40	5	10	20	40
033^3	< 1	< 1	< 1	< 1	< 1	< 1	< 1	< 1	1.	< 1	< 1	< 1
065^3	12.	8.	5.	6.	10.	5.	5.	6.	45.	6.	5.	7.
129^3	436.	282.	233.	172.	327.	281.	132.	175.	-8.	182.	143.	179.
257^3	-8.	8598.	6301.	NA	9610.	3662.	2757.	NA	-3.	4539.	2452.	NA
	Number of iterations											
	5	10	20	40	5	10	20	40	5	10	20	40
033^3	65	40	40	40	60	40	40	40	260	50	40	40
065^3	205	130	80	80	165	90	80	80	860	100	80	80
129^3	715	410	260	160	500	410	140	160	2000	250	160	160
257^3	2000	1440	820	NA	1695	570	340	NA	2000	690	280	NA

Table 8. Comparison of the effectiveness of enrichment on the diffusion test problem. The diffusion coefficients used in the study are $\epsilon_x = 10000$; $\epsilon_y = 100$; $\epsilon_z = 1$. All test parameters are identical to those used in the uniform grid case (see table 8).

Enrichment	10000:100:1 Aspect Grid: $ILU(0)$											
Vectors	Enrichment vectors : 0				Enrichment vectors : 2				Enrichment vectors : 4			
	CPU-Time											
Grid	5	10	20	40	5	10	20	40	5	10	20	40
033^3	< 1	< 1	1.	1.	< 1	< 1	1.	1.	1.	< 1	1.	1.
065^3	2.	4.	2.	4.	11.	5.	3.	4.	39.	7.	3.	8.
129^3	63.	57.	68.	100.	138.	75.	76.	105.	794.	102.	89.	112.
257^3	2124.	1283.	1355.	NA	4139.	1759.	1526.	NA	-5.	2430.	1704.	NA
	Number of iterations											
	5	10	20	40	5	10	20	40	5	10	20	40
033^3	15	10	20	40	15	10	20	40	35	10	20	40
065^3	50	50	60	40	85	60	60	40	430	70	60	40
129^3	135	100	100	120	250	120	100	120	1605	160	100	120
257^3	500	260	200	NA	875	320	220	NA	2000	420	220	NA
Enrichment	10000:100:1 Aspect Grid: $ILU(1)$											
Vectors	Enrichment vectors : 0				Enrichment vectors : 2				Enrichment vectors : 4			
	CPU-Time											
	5	10	20	40	5	10	20	40	5	10	20	40
033^3	< 1	< 1	< 1	< 1	< 1	< 1	3.	< 1	< 1	< 1	< 1	< 1
065^3	0.	2.	0.	3.	4.	0.	0.	3.	19.	0.	0.	3.
129^3	66.	59.	68.	98.	133.	75.	77.	99.	717.	99.	85.	105.
257^3	2114.	1332.	1335.	NA	3779.	1734.	1485.	NA	-6.	2285.	1631.	NA
	Number of iterations											
	5	10	20	40	5	10	20	40	5	10	20	40
033^3	20	20	20	20	35	20	50	20	105	20	20	20
065^3	30	40	40	40	75	20	40	40	350	20	40	40
129^3	110	90	80	80	205	110	80	80	1280	140	100	80
257^3	400	230	180	NA	665	270	180	NA	2000	350	200	NA

Table 9. Comparison of the effectiveness of enrichment on the diffusion test problem. The diffusion coefficients used in the study are $\epsilon_x = 10000$; $\epsilon_y = 100$; $\epsilon_z = 1$. All test parameters are identical to those used in the uniform grid case (see table 8).

Enrichment Vectors	For every new problem: discard $\mathcal{A}\ S_m{}^{j-1} = V_{m+1}{}^{j-1}\,\bar{\mathcal{H}}_m{}^{j-1}$					For every new problem, pre-enrich: retain $\mathcal{A}\ S_m{}^{j-1} = V_{m+1}{}^{j-1}\,\bar{\mathcal{H}}_m{}^{j-1}$				
	Preconditioner: $ILU(2)$									
	Maximum Krylov subspace dimension									
	20	25	30	35	40	20	25	30	35	40
0	2366.	995.	695.	530.	480.	2497.	1003.	727.	563.	487.
2	685.	545.	482.	422.	433.	684.	517.	459.	413.	394.
4	532.	469.	417.	412.	388.	583.	507.	441.	399.	393.
6	493.	439.	412.	394.	381.	580.	466.	426.	400.	374.
8	469.	419.	395.	382.	398.	507.	421.	381.	391.	379.
10	995.	695.	530.	480.	386.	1003.	727.	563.	487.	377.
	Preconditioner: $ILU(3)$									
	Maximum Krylov subspace dimension									
	20	25	30	35	40	20	25	30	35	40
0	559.	478.	428.	403.	384.	592.	488.	452.	404.	396.
2	498.	423.	411.	405.	385.	506.	443.	411.	404.	385.
4	384.	380.	378.	353.	376.	381.	367.	356.	338.	337.
6	381.	363.	358.	345.	369.	378.	353.	342.	327.	350.
8	372.	360.	351.	336.	361.	370.	341.	337.	314.	338.
10	478.	428.	403.	384.	356.	488.	452.	404.	396.	337.
	Preconditioner: $ILU(4)$									
	Maximum Krylov subspace dimension									
	20	25	30	35	40	20	25	30	35	40
0	339.	316.	337.	346.	316.	341.	318.	340.	348.	331.
2	327.	319.	334.	344.	315.	323.	316.	331.	342.	351.
4	320.	316.	326.	338.	311.	337.	317.	319.	332.	343.
6	319.	310.	318.	329.	307.	337.	311.	305.	321.	346.
8	324.	315.	305.	324.	302.	342.	328.	305.	319.	345.
10	316.	337.	346.	316.	299.	318.	340.	348.	331.	367.

Table 10. Convergence study for unsteady 2D Turbulent NS equations with $k = 0, 2, 4, 6, 8, 10$ harmonic Ritz vectors to enrich eigenspace. The eigen-selection routine sorts the Ritz values based on the formula $|\zeta_j| = [\frac{\sqrt{((-0.25)-\theta_r)_j^2 + (\theta_i)_j^2}}{\sqrt{(1-\theta_r)_j^2 + (\theta_i)_j^2}}]$, and chooses the k smallest values of ζ_j. The data presented is the CPU time required to reach the specified tolerance.

Enrichment eigenvalue	Total time
-0.163, 0.0	663.5
-0.40, 0.0	698.2
-1.26, 0.0	852.6
-2.50, 0.0	869.5
0.16, 0.0	886.7
0.18, 0.0	901.1
-23.28, 0.0	958.9
122.63, 0.0	968.4
13.29, 0.0	1016.8

Table 11. Influence on CPU time when the Krylov basis is enriched with one eigenvector. Smaller times indicate eigenvectors that, when removed, greatly improved the convergence rate.

IRK	Iteration	Initial reduction
2	1	Not applicable
2	2	0.97
2	3	0.99
2	4	0.46
2	5	0.88
2	6	0.98
3	1	Not applicable
3	2	0.93
3	3	0.82
4	1	Not applicable
4	2	0.92
4	3	0.90
4	4	0.92

Table 12. Initial reduction of residual, based on previous eigenvector. Greater values of reduction indicate greater linear dependence on the nonlinear iterations.

Enrichment	Maximum Krylov subspace dimension									
Vectors	$\|\zeta_j\| = [\sqrt{(\theta_r)_j^2 + (\theta_i)_j^2}]$					$\|\zeta_j\| = [\frac{(\theta_r)_j}{\sqrt{(1-\theta_r)_j^2 + (\theta_i)_j^2}}]$				
	20	25	30	35	40	20	25	30	35	40
0	2358.	997.	696.	528.	477.	2361.	990.	696.	528.	477.
2	0.	710.	549.	487.	452.	1479.	864.	620.	499.	447.
4	1022.	675.	540.	482.	443.	797.	583.	487.	449.	420.
6	1067.	646.	539.	466.	439.	497.	445.	407.	394.	371.
8	0.	711.	556.	467.	433.	477.	423.	388.	382.	398.
10	997.	696.	528.	477.	421.	990.	696.	528.	477.	396.
	$\|\zeta_j\| = [\frac{1}{\sqrt{(1-\theta_r)_j^2 + (\theta_i)_j^2}}]$					$\|\zeta_j\| = [\frac{\sqrt{((-0.25)-\theta_r)_j^2 + (\theta_i)_j^2}}{\sqrt{(1-\theta_r)_j^2 + (\theta_i)_j^2}}]$				
	20	25	30	35	40	20	25	30	35	40
0	2397.	1008.	716.	535.	483.	2366.	995.	695.	530.	480.
2	2173.	928.	695.	527.	480.	685.	545.	482.	422.	433.
4	0.	967.	677.	546.	470.	532.	469.	417.	412.	388.
6	1853.	870.	648.	507.	466.	493.	439.	412.	394.	381.
8	0.	887.	640.	508.	467.	469.	419.	395.	382.	398.
10	1008.	716.	535.	483.	452.	995.	695.	530.	480.	386.

Table 13. Convergence study for unsteady 2D Turbulent NS equations with $ILU(2)$ as preconditioner and $k = 0, 2, 4, 6, 8, 10$ harmonic Ritz vectors to enrich eigenspace. The eigen-selection routine sorts the Ritz values based one of four different formulas. Presented is the CPU time required to reach the specified tolerance.

Enrichment	Cumulative CPU time									
vectors	Harmonic Ritz					Conventional Ritz				
	Maximum Krylov subspace dimension									
	20	25	30	35	40	20	25	30	35	40
0	2366.	995.	695.	530.	480.	2393.	994.	700.	531.	478.
2	685.	545.	482.	422.	433.	751.	538.	481.	422.	432.
4	532.	469.	417.	412.	388.	616.	470.	419.	410.	387.
6	493.	439.	412.	394.	381.	518.	452.	418.	401.	398.
8	469.	419.	395.	382.	398.	478.	431.	394.	382.	397.
10	995.	695.	530.	480.	386.	994.	700.	531.	478.	389.

Table 14. Convergence study for unsteady 2D Turbulent NS equations with $ILU(2)$ as preconditioner and $k = 0, 2, 4, 6, 8, 10$ harmonic Ritz or conventional Ritz vectors to enrich eigenspace. The eigen-selection routine sorts the Ritz values based on the formula $|\zeta_j| = [\frac{\sqrt{((-0.25)-\theta_r)_j^2 + (\theta_i)_j^2}}{\sqrt{(1-\theta_r)_j^2 + (\theta_i)_j^2}}]$.

Enrichment	BL-RCM: 13 Linear Solutions					RCM: 11 Linear Solutions				
vectors	Preconditioner: $ILU(2)$									
	Maximum Krylov subspace dimension									
	20	25	30	35	40	20	25	30	35	40
0	2393.	994.	700.	531.	478.	0.	0.	0.	0.	0.
2	751.	538.	481.	422.	432.	0.	0.	0.	0.	0.
4	616.	470.	419.	410.	387.	0.	0.	0.	0.	0.
6	518.	452.	418.	401.	398.	0.	0.	0.	0.	2718.
8	478.	431.	394.	382.	397.	0.	2385.	1307.	1105.	944.
10	994.	700.	531.	478.	389.	0.	0.	0.	0.	777.
	Preconditioner: $ILU(3)$									
	Maximum Krylov subspace dimension									
	20	25	30	35	40	20	25	30	35	40
0	561.	480.	428.	403.	384.	204.	190.	200.	233.	221.
2	509.	423.	411.	405.	385.	201.	180.	199.	232.	220.
4	394.	381.	377.	353.	374.	191.	188.	195.	228.	218.
6	395.	368.	373.	345.	368.	188.	196.	192.	225.	216.
8	384.	361.	358.	336.	362.	198.	207.	187.	221.	213.
10	480.	428.	403.	384.	355.	190.	200.	233.	221.	210.
	Preconditioner: $ILU(4)$									
	Maximum Krylov subspace dimension									
	20	25	30	35	40	20	25	30	35	40
0	339.	316.	337.	346.	316.	235.	200.	236.	228.	167.
2	327.	320.	335.	344.	315.	212.	198.	232.	226.	167.
4	322.	323.	325.	338.	311.	220.	192.	229.	224.	167.
6	340.	310.	317.	331.	307.	210.	188.	224.	220.	167.
8	327.	306.	305.	322.	302.	197.	182.	218.	217.	175.
10	316.	337.	346.	316.	299.	200.	236.	228.	167.	167.

Table 15. Convergence study for unsteady 2D Turbulent NS equations with $k = 0, 2, 4, 6, 8, 10$, conventional Ritz vectors to enrich eigenspace. The eigen-selection routine sorts the Ritz values based on the formula $|\zeta_j| = [\frac{\sqrt{((-0.25)-\theta_r)_j^2+(\theta_i)_j^2}}{\sqrt{(1-\theta_r)_j^2+(\theta_i)_j^2}}]$. Two grid orderings are used in study.

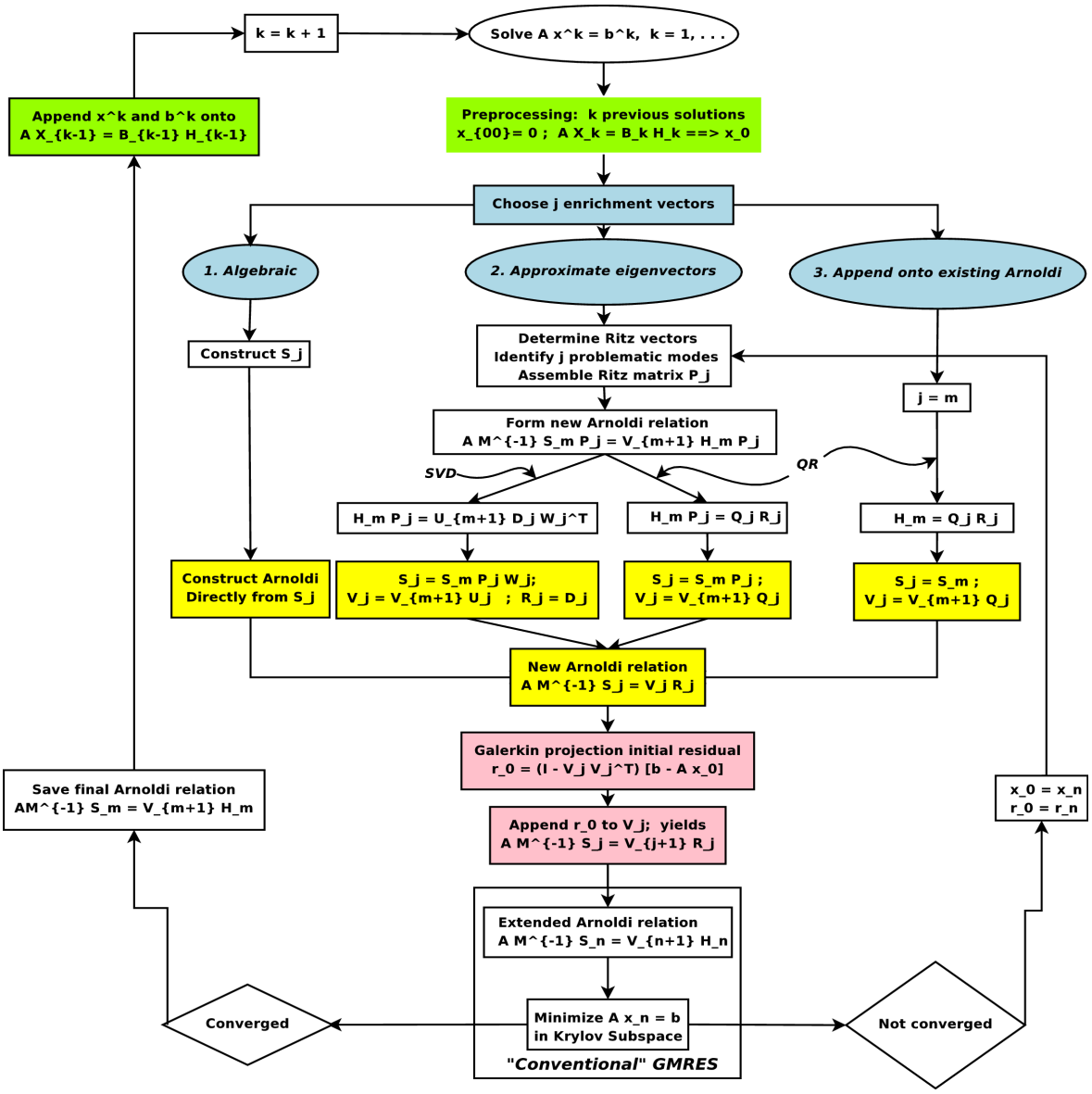


Figure 1. Schematic of the proposed enrichment algorithm.

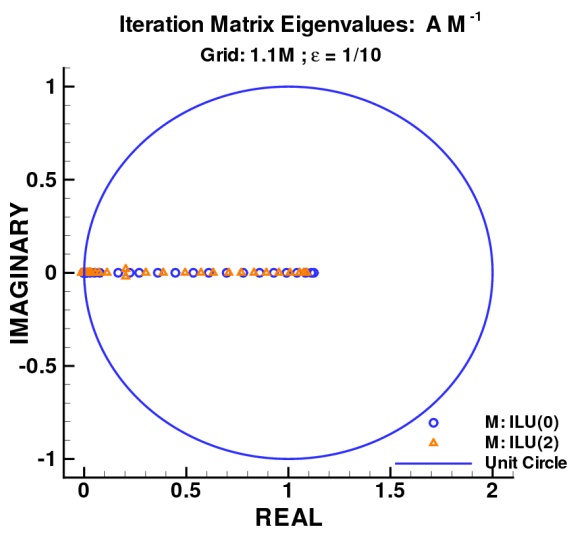


Figure 2. Advection-diffusion eigenvalues for the parameter $\epsilon = \frac{1}{10}$. The two preconditioners used in the study are $ILU(0)$ and $ILU(2)$.

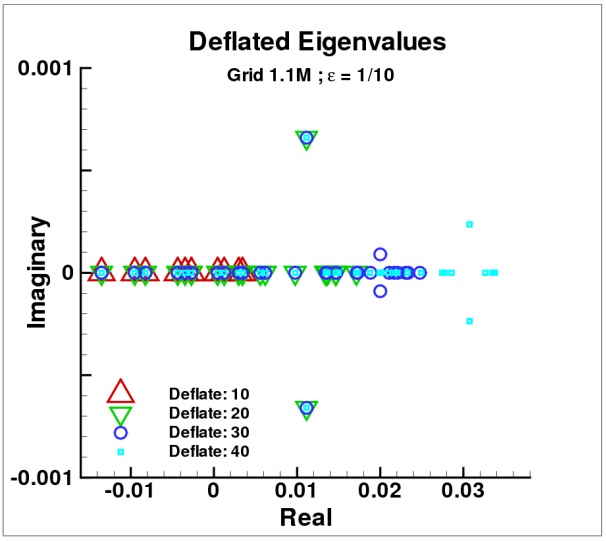


Figure 3. Comparison of eigenvalues obtained using 10, 20, 30, and 40 enrichment vectors. The Krylov dimension is 60 in all cases. Preconditioner is $ILU(1)$.

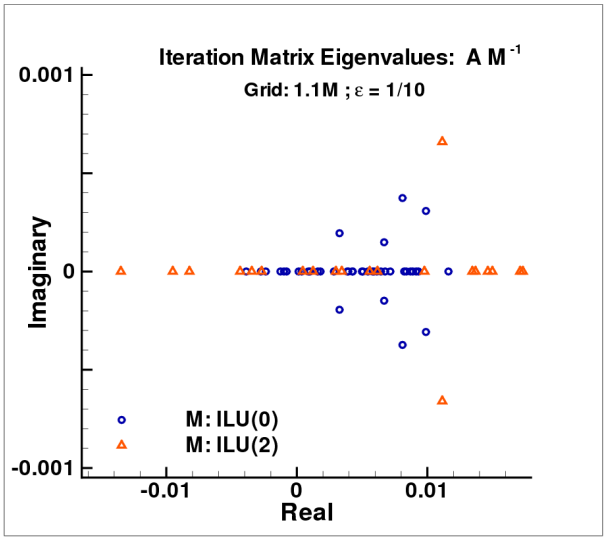


Figure 4. Advection-diffusion eigenvalues located near the origin for the parameter $\epsilon = \frac{1}{10}$ with two preconditioners $ILU(0)$ and $ILU(2)$.

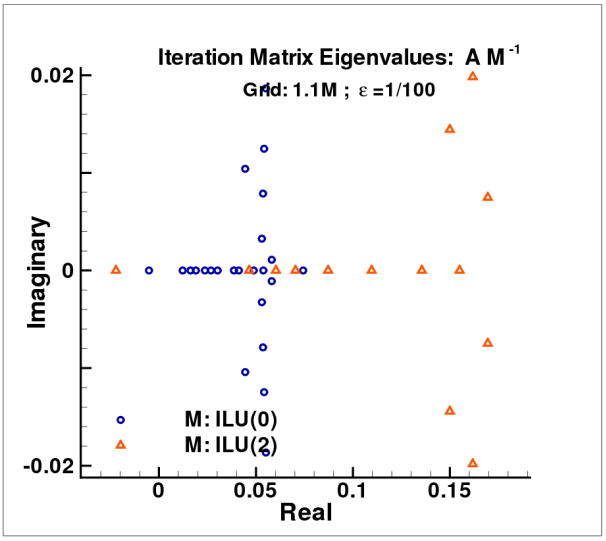


Figure 5. Advection-diffusion eigenvalues located near the origin for the parameter $\epsilon = \frac{1}{100}$ with two preconditioners $ILU(0)$ and $ILU(2)$.

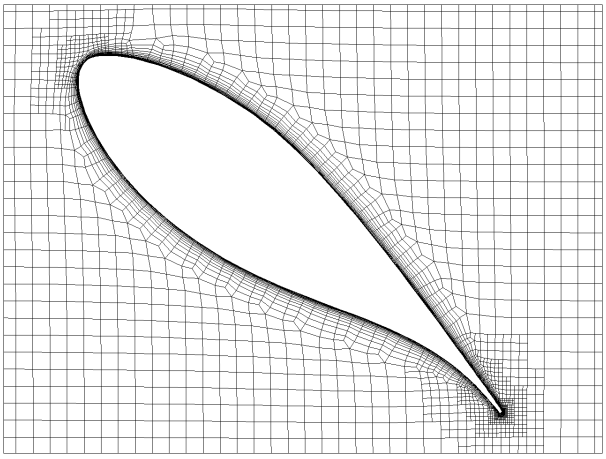


Figure 6. Close-up grid of airfoil.

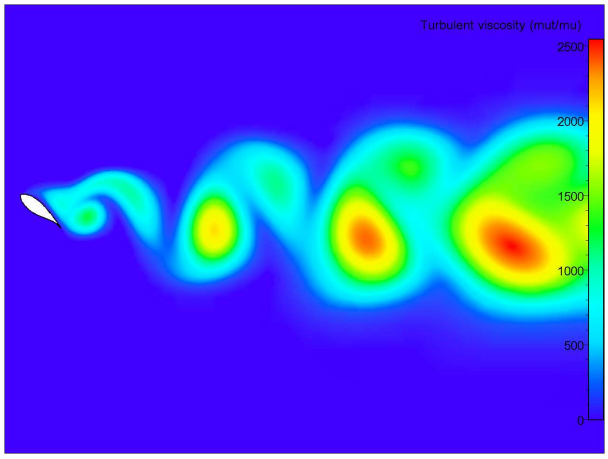


Figure 7. Turbulent viscosity.

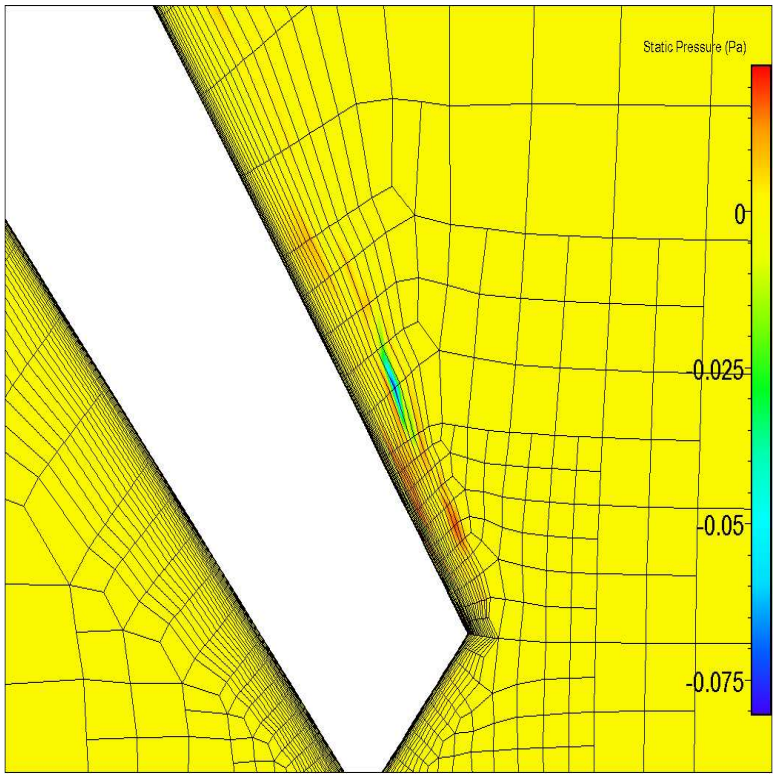


Figure 8. Plot of the dominant problematic eigenvalue; the eigenvalue is (-0.16, 0,0) in the complex plane.

REPORT DOCUMENTATION PAGE	Form Approved OMB No. 0704–0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
01-01-2010	Technical Memorandum	10/2009–12/2009

4. TITLE AND SUBTITLE
A General Algorithm for Reusing Krylov Subspace Information.
I. Unsteady Navier-Stokes

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Mark H. Carpenter, C. Vuik, Peter Lucas, Martin van Gijzen, Hester Bijl

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER
599489.02.07.07.03.13.01

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
NASA Langley Research Center
Hampton, Virginia 23681-2199

8. PERFORMING ORGANIZATION REPORT NUMBER
L–19793

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
National Aeronautics and Space Administration
Washington, DC 20546-0001

10. SPONSOR/MONITOR'S ACRONYM(S)
NASA

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
NASA/TM–2010–216190

12. DISTRIBUTION/AVAILABILITY STATEMENT
Unclassified-Unlimited
Subject Category 64
Availability: NASA CASI (443) 757-5802

13. SUPPLEMENTARY NOTES
An electronic version can be found at http://ntrs.nasa.gov.

14. ABSTRACT

A general algorithm is developed that reuses available information to accelerate the iterative convergence of linear systems with multiple right-hand sides $\mathcal{A}\,\mathbf{x} = \mathbf{b}^i$, which are commonly encountered in steady or unsteady simulations of nonlinear equations. The algorithm is based on the classical GMRES algorithm with eigenvector enrichment but also includes a Galerkin projection preprocessing step and several novel Krylov subspace reuse strategies. The new approach is applied to a set of test problems, including an unsteady turbulent airfoil, and is shown in some cases to provide significant improvement in computational efficiency relative to baseline approaches.

15. SUBJECT TERMS
Krylov, GMRES, deflation, enrichment, numerical stability, Galerkin Projection

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			STI Help Desk (email: help@sti.nasa.gov)
U	U	U	UU	52	19b. TELEPHONE NUMBER *(Include area code)* (443) 757-5802