New Approaches to Large Scale Eigenanalysis

D.C. Sorensen

CRPC-TR97776
February 1997

Center for Research on Parallel Computation
Rice University
6100 South Main Street
CRPC - MS 41
Houston, TX 77005

Submitted September 1998; In "Computational Science for the 21st Century", Bristeau, Etgen, Fitzgibbon, Lions, Periaux and Wheeler, eds., 62-71, John Wiley and Sons Ltd, Chichester, England, 1977

New Approaches to Large Scale Eigenanalysis

D.C. Sorensen
Rice University

1 INTRODUCTION

The past few years have seen significant advances in numerical techniques to compute partial eigendecompositions of large matrices. These new approaches have led to critical advances in several application areas including computational chemistry, semi-conductor laser design, linear stability analysis, and reduced basis techniques for large state space control systems.

This paper will survey some of these new techniques. In particular, we shall discuss the implicitly restarted Arnoldi method (IRAM) which is the foundation for the eigenvalue software package ARPACK[10]. This package has been used extensively in many application areas that require the solution of large scale symmetric and nonsymmetric (generalized) eigenvalue problems. A brief introduction to Krylov subspace projection is given and the Lanczos/Arnoldi factorization is introduced. Implicit restarting is presented as a means for controlling computational cost, maintaining numerical accuracy, avoiding "ghost" eigenvalues, and computing selected eigenvalues of specific interest. The large scale eigenvalue software ARPACK that is based on this computational framework is discussed. Finally, new approaches currently under development are presented that promise to address some of the most difficult challenges that remain in this active research area.

2 PROJECTION METHODS AND THE LANCZOS/ARNOLDI PROCESS

The power method is surely the most basic of methods for eigenvalue calculations but it has severe limitations. The successive vectors produced by a power method may contain considerable information along eigenvector directions other than the dominant one, but the power method systematically ignores this additional information. Fortunately, more sophisticated techniques may be employed to extract it.

It is natural to formally consider the linear span of the vectors produced by the power iteration and to attempt to extract additional information by formulating approximate eigenvectors from this subspace. This *Krylov* subspace

$$\mathcal{K}_k(\mathbf{A}, \mathbf{v}_1) = \text{Span } \{\mathbf{v}_1, \mathbf{A}\mathbf{v}_1, \mathbf{A}^2\mathbf{v}_1, \ldots, \mathbf{A}^{k-1}\mathbf{v}_1\}$$

is central to the theory. Its elements are of the form $p(\mathbf{A})v_1$ where p is a polynomial and this leads to deep connections with polynomial approximation theory, orthogonal polynomials, theory of moments and Padé approximation. Optimal approximate eigenpairs, called *Ritz pairs*, may be obtained by imposing a Galerkin condition: A vector $\mathbf{x} \in \mathcal{K}_k(\mathbf{A}, \mathbf{v}_1)$ is called a *Ritz vector* with corresponding *Ritz value* θ if the Galerkin condition

$$< \mathbf{w}, \mathbf{A}\mathbf{x} - \mathbf{x}\theta >= 0 \ , \quad \text{for all} \quad \mathbf{w} \in \mathcal{K}_k(\mathbf{A}, \mathbf{v}_1)$$

is satisfied. Various optimality conditions follow from this definition that are important for understanding the approximation properties of the *Ritz pair* $(\mathbf{x}, \theta)$. This definition is equivalent to projection and many of these properties may be developed with co-ordinate free arguments but it is convenient to consider a specific basis in a computational setting.

2.1 *THE ARNOLDI FACTORIZATION*

For computational purposes, there is an efficient way to produce a particular ortho-normal basis for $\mathcal{K}_k(\mathbf{A}, \mathbf{v}_1)$. This takes the form of a truncated orthogonal similarity reduction of A to condensed form.

Definition : If $\mathbf{A} \in \mathbf{C}^{n\times n}$ then a relation of the form

$$\mathbf{A}\mathbf{V}_k = \mathbf{V}_k\mathbf{H}_k + \mathbf{f}_k\mathbf{e}_k^T \tag{1}$$

where $\mathbf{V}_k \in \mathbf{C}^{n\times k}$ has orthonormal columns, $\mathbf{V}_k^H\mathbf{f}_k = 0$ and $\mathbf{H}_k \in \mathbf{C}^{k\times k}$ is upper Hessenberg with non-negative subdiagonal elements is called a *k-step Arnoldi Factorization* of $\mathbf{A}$. If $\mathbf{A}$ is Hermitian then $\mathbf{H}_k$ is real, symmetric and tridiagonal and the relation is called a *k-step Lanczos Factorization* of $\mathbf{A}$. The columns of $\mathbf{V}_k$ are referred to as the *Arnoldi vectors* or *Lanczos vectors*, respectively.

This factorization may be used to obtain approximate eigenvalues and eigenvectors of $\mathbf{A}$. It is easily shown that the Ritz pairs defined previously by the Galerkin condition are immediately available from the eigenpairs of the small projected matrix $\mathbf{H}_k$.

If $\mathbf{H}_k\mathbf{s} = \mathbf{s}\theta$, then the vector $\mathbf{x} = \mathbf{V}_k\mathbf{s}$ satisfies

$$||\mathbf{A}\mathbf{x} - \mathbf{x}\theta|| = ||(\mathbf{A}\mathbf{V}_k - \mathbf{V}_k\mathbf{H}_k)\mathbf{s}|| = |\beta_k\mathbf{e}_k^T\mathbf{s}|.$$

The number $|\beta_k\mathbf{e}_k^T\mathbf{s}|$ is called the *Ritz estimate* for the Ritz pair $(\mathbf{x}, \theta)$ as an approximate eigenpair for $\mathbf{A}$. When $\mathbf{A}$ is Hermitian, the Ritz estimate may be used to provide computable rigorous bounds on the accuracy of the eigenvalues of $\mathbf{H}_k$ as approximations to eigenvalues [12] of $\mathbf{A}$. When $\mathbf{A}$ is non-Hermitian the possibility of non-normality precludes such bounds. However, in either case, if $\mathbf{f}_k = 0$ these the Ritz pairs become exact eigenpairs of $\mathbf{A}$.

Some of the optimality conditions alluded to in the previous section are revealed through this factorization. A particularly useful fact for establishing error estimates and asymptotic behavior is

$$\mathbf{f}_k = \frac{p_k(\mathbf{A})\mathbf{v}_1}{||p_{k-1}(\mathbf{A})\mathbf{v}_1||} \quad \text{where} \quad p_k(\lambda) = det(\lambda\mathbf{I} - \mathbf{H}_k) \tag{2}$$

and $p = p_k$ minimizes $||p(\mathbf{A})\mathbf{v}_1||$ over all monic polynomials p of degree k [15, 17].

The k-step factorization may be advanced one step at the cost of a (sparse) matrix-vector product involving $\mathbf{A}$ and two dense matrix vector products involving $\mathbf{V}_k^H$ and $\mathbf{V}_k$. The explicit steps needed to form a k-Step Arnoldi Factorization are listed in Algorithm 2 shown in Figure 1. In exact arithmetic, the columns of $\mathbf{V}_j$ form an orthonormal basis for the Krylov subspace and $\mathbf{H}_j$ is the orthogonal projection of $\mathbf{A}$ onto this space. In finite precision arithmetic, care must be taken to assure that the computed vectors are orthogonal to working precision. The method proposed by Daniel, Gragg, Kaufman and Stewart (DGKS) in [1] provides an excellent means to construct a vector $\mathbf{f}_{j+1}$ that is numerically orthogonal to $\mathbf{V}_{j+1}$.

Step (3.4) in Algorithm 1 is nothing more than Classical Gram Schmidt (CGS) orthogonalization of the vector $\mathbf{A}\mathbf{v}_{j+1}$ with respect to the columns of $\mathbf{V}_{j+1}$. The dense matrix-vector products for this step may be formulated using Level 2 BLAS. This allows a very efficient and portable parallel implementation [11] with far better performance than is possible with Level 1 BLAS [3, 4]. In the Arnoldi process, a modified Gram-Schmidt algorithm (MGS) is often recommended for this calculation. However, in the Arnoldi context, MGS cannot be formulated with Level 2 BLAS

Algorithm 1: The k-Step Arnoldi Factorization
Input: $(\mathbf{A}, \mathbf{v}_1)$
Output: $(\mathbf{V}_k, \mathbf{H}_k, \mathbf{f}_k)$ such that $\mathbf{AV}_k = \mathbf{V}_k\mathbf{H}_k + \mathbf{f}_k\mathbf{e}_k^T$,
$\mathbf{V}_k^H\mathbf{V}_k = \mathbf{I}$, $\mathbf{V}_k^H\mathbf{f}_k = \mathbf{0}$ and $\mathbf{H}_k$ upper Hessenberg.

1. Put $\mathbf{v}_1 = \mathbf{v}/\|\mathbf{v}_1\|$; $\mathbf{w} = \mathbf{Av}_1$; $\alpha_1 = \mathbf{v}_1^H\mathbf{w}$;
2. Put $\mathbf{f}_1 \leftarrow \mathbf{w} - \mathbf{v}_1\alpha_1$; $\mathbf{V}_1 \leftarrow (\mathbf{v}_1)$; $\mathbf{H}_1 \leftarrow (\alpha_1)$;
3. **for** $j = 1, 2, \cdots, k-1$,
   3.1. $\beta_j = \|\mathbf{f}_j\|$; $\mathbf{v}_{j+1} \leftarrow \mathbf{f}_j/\beta_j$;
   3.2. $\mathbf{V}_{j+1} \leftarrow (\mathbf{V}_j, \mathbf{v}_{j+1})$; $\hat{\mathbf{H}}_j \leftarrow \begin{pmatrix} \mathbf{H}_j \\ \beta_j\mathbf{e}_j^T \end{pmatrix}$;
   3.3. $\mathbf{z} \leftarrow \mathbf{Av}_{j+1}$;
   3.4. $\mathbf{h} \leftarrow \mathbf{V}_{j+1}^H\mathbf{z}$; $\mathbf{f}_{j+1} \leftarrow \mathbf{z} - \mathbf{V}_{j+1}\mathbf{h}$;
   3.5. $\mathbf{H}_{j+1} \leftarrow (\hat{\mathbf{H}}_j, \mathbf{h})$;
4. **end**;

Figure 1: The k-Step Arnoldi Factorization

and does not attain the required level of numerical orthogonality that CGS with DGKS correction achieves. This is particularly important with respect to restarting.

3 IMPLICIT RESTARTING

The Arnoldi process becomes prohibitively expensive as soon as k becomes large. However, in practice, many steps may be required before accurate approximations appear. Restarting was proposed very early as a means to overcome these problems [15]. One idea is to fix the length of the factorization at m steps say, and repeatedly replace the starting vector $\mathbf{v}_1 \leftarrow \psi(\mathbf{A})\mathbf{v}_1$ after every m steps. The polynomial ψ is constructed to filter out "unwanted" components in the eigenvector expansion of $\mathbf{v}_1$ based upon information learned about the spectrum of $\mathbf{A}$ as the iteration proceeds.

There is another approach to restarting that offers a more efficient and numerically stable formulation. This approach, called *implicit restarting*, is a technique for combining the implicitly shifted QR-mechanism with an m-step Arnoldi or Lanczos factorization to obtain a truncated form of the implicitly shifted QR-iteration [17]. The numerical difficulties and storage problems normally associated with Arnoldi and Lanczos processes are avoided. There are no problems with "ghost" eigenvalue approximations since the basis vectors are maintained to full numerical orthogonality. The algorithm is capable of computing a few ($k < m$) eigenvalues with user specified features such as largest real part or largest magnitude using $m \cdot n + \mathcal{O}(m^2)$storage. No auxiliary storage is required. The computed Schur basis vectors for the desired k-dimensional eigen-space are numerically orthogonal to working precision. The suitability of this method for the development of mathematical software stems from this concise and automatic treatment of the primary difficulties with the Arnoldi/Lanczos process.

Implicit restarting provides a means to extract interesting information from large Krylov subspaces while avoiding the storage and numerical difficulties associated with the standard approach. It does this by continually compressing the interesting information into a fixed size k-dimensional subspace. This is accomplished through the implicitly shifted QR-mechanism. An Arnoldi factor-

ization of length $m = k + p$

$$\mathbf{A}\mathbf{V}_m = \mathbf{V}_m\mathbf{H}_m + \mathbf{f}_m\mathbf{e}_m^T, \tag{3}$$

is compressed to a factorization of length k that retains the eigen-information of interest. This is accomplished using QR-steps to apply p shifts implicitly. The first stage of this shift process results in

$$\mathbf{A}\mathbf{V}_m^+ = \mathbf{V}_m^+\mathbf{H}_m^+ + \mathbf{f}_m\mathbf{e}_m^T\mathbf{Q}, \tag{4}$$

where $\mathbf{V}_m^+ = \mathbf{V}_m\mathbf{Q}$, $\mathbf{H}_m^+ = \mathbf{Q}^H\mathbf{H}_m\mathbf{Q}$, and $\mathbf{Q} = \mathbf{Q}_1\mathbf{Q}_2\cdots\mathbf{Q}_p$, with $\mathbf{Q}_j$ the orthogonal matrix associated with the shift μ_j. Because of the Hessenerg structure of the matrices $\mathbf{Q}_j$, it turns out that the first $k-1$ entries of the vector $\mathbf{e}_m^T\mathbf{Q}$ are zero (i.e. $\mathbf{e}_m^T\mathbf{Q} = (\gamma\mathbf{e}_k^T, \hat{\mathbf{q}}^H)$). This implies that the leading k columns in equation (4) remain in an Arnoldi relation. Equating the first k columns on both sides of (4) provides an updated $k-$step Arnoldi factorization

$$\mathbf{A}\mathbf{V}_k^+ = \mathbf{V}_k^+\mathbf{H}_k^+ + \mathbf{f}_k^+\mathbf{e}_k^T, \tag{5}$$

with an updated residual of the form $\mathbf{f}_k^+ = \mathbf{V}_m^+\mathbf{e}_{k+1}\hat{\beta}_k + \mathbf{f}_m\gamma$. Using this as a starting point it is possible to apply p additional steps of the Arnoldi process to return to the original m-step form. Each m-step factorization is obtained at a cost of only p additional matrix-vector products involving $\mathbf{A}$ together with the the arithmetic required to form $\mathbf{V}_m^+ = \mathbf{V}_m\mathbf{Q}$.

Each of these shift cycles results in the implicit application of a polynomial in $\mathbf{A}$ of degree p to the starting vector.

$$\mathbf{v}_1 \leftarrow \psi(\mathbf{A})\mathbf{v}_1 \quad \text{with} \quad \psi(\lambda) = \prod_1^p(\lambda - \mu_j). \tag{6}$$

The roots of this polynomial are the shifts used in the QR-process and these may be selected to filter unwanted information from the starting vector and hence from the Arnoldi factorization. Full details including a convergence analysis may be found in [17]. Further implementation details and development of appropriate deflation techniques may be found in [9].

This repeated updating of the starting vector $\mathbf{v}_1$ through implicit restarting is designed to enhance the components of this vector in the directions of the wanted eigenvectors and damp its components in the unwanted directions. If $\mathbf{v}_1 = \sum_{j=1}^n \mathbf{x}_j\gamma_j$ has an expansion as a linear combination of eigenvectors $\{\mathbf{x}_j\}$ of $\mathbf{A}$, the polynomial restart vector $\mathbf{v}_1^+$ satisfies

$$\mathbf{v}_1^+ \equiv \psi(\mathbf{A})\mathbf{v}_1 = \sum_{j=1}^n \mathbf{x}_j\,\psi(\lambda_j)\gamma_j.$$

If ψ is repeatedly constructed to take large values at the "wanted" eigenvalues $\{\lambda_1, \lambda_2, \cdots, \lambda_k\}$ and relatively small values (in magnitude) on the remaining "unwanted" then eventually all unwanted components are damped to zero and the desired eigenvalues will be found.

This final statement is justified by the fact that if the starting vector $\mathbf{v}_1 = \sum_{j=1}^k \mathbf{x}_j\gamma_j$ where $\mathbf{A}\mathbf{x}_j = \mathbf{x}_j\lambda_j$, then $\mathbf{f}_k = \mathbf{0}$. This follows easily from the optimality condition (2). Moreover, in this event $\mathbf{A}\mathbf{V}_k = \mathbf{V}_k\mathbf{H}_k$. Hence, $\mathbf{V}_k$ will provide an orthonormal basis for the invariant subspace $\mathcal{S} \equiv \mathit{Range}(\mathbf{V}_k)$. and again from (2), $\sigma(\mathbf{H}_k) = \{\lambda_1, \lambda_2, \cdots, \lambda_k\}$.

4 IRA and ARPACK

A large scale eigenvalue software package called ARPACK [10] has been developed in Fortran 77 based upon IRAM for nonsymmetric problems and IRLM for symmetric (Hermition) problems. The package is designed to compute a few eigenvalues and corresponding eigenvectors of a general $n \times n$ matrix $\mathbf{A}$. It is most appropriate for large sparse or structured matrices $\mathbf{A}$, where structured

Algorithm 2: (IRAM) Implicitly Restarted Arnoldi Method

Input: $(\mathbf{A}, \mathbf{V}, \mathbf{H}, \mathbf{f})$ with $\mathbf{AV}_m = \mathbf{V}_m\mathbf{H}_m + \mathbf{f}_m\mathbf{e}_m^T$, an m-Step Arnoldi Factorization;
Output: $(\mathbf{V}_k, \mathbf{H}_k)$ such that $\mathbf{AV}_k = \mathbf{V}_k\mathbf{H}_k$, $\mathbf{V}_k^H\mathbf{V}_k = \mathbf{I}$ and $\mathbf{H}_k$ is upper triangular.

1. **for** $\ell = 1, 2, 3, \ldots$ until *convergence*
   1.1. Compute $\sigma(\mathbf{H}_m)$ and select set of p shifts $\mu_1, \mu_2, \ldots \mu_p$
   based upon $\sigma(\mathbf{H}_m)$ or perhaps other information;
   1.2. $\mathbf{q}^T \leftarrow \mathbf{e}_m^T$;
   1.3. **for** $j = 1, 2, \ldots, p$,
   1.3.1. Factor $[\mathbf{Q}, \mathbf{R}] = qr(\mathbf{H}_m - \mu_j\mathbf{I})$;
   1.3.2. $\mathbf{H}_m \leftarrow \mathbf{Q}^H\mathbf{H}_m\mathbf{Q}$; $\mathbf{V}_m \leftarrow \mathbf{V}_m\mathbf{Q}$;
   1.3.3. $\mathbf{q} \leftarrow \mathbf{q}^H\mathbf{Q}$;
   1.4 **end**;
   1.5. $\mathbf{f}_k \leftarrow \mathbf{v}_{k+1}\hat{\beta}_k + \mathbf{f}_m\gamma$; $\mathbf{V}_k \leftarrow \mathbf{V}_m(1:n, 1:k)$; $\mathbf{H}_k \leftarrow \mathbf{H}_m(1:k, 1:k)$;
   1.6. Beginning with the k-step Arnoldi factorization
   $\mathbf{AV}_k = \mathbf{V}_k\mathbf{H}_k + \mathbf{f}_k\mathbf{e}_k^T$,
   apply p additional Arnoldi steps to get a new m-step Arnoldi factorization
   $\mathbf{AV}_m = \mathbf{V}_m\mathbf{H}_m + \mathbf{f}_m\mathbf{e}_m^T$.
2. **end**;

Figure 2: An Implicitly Restarted Arnoldi Method.

means that a matrix-vector product $\mathbf{w} \leftarrow \mathbf{Av}$ requires order n rather than the usual order n^2 floating point operations. For many standard problems, a matrix factorization is not required. Only the action of the matrix on a vector is needed.

ARPACK software is capable of solving large scale symmetric, nonsymmetric, and generalized eigenproblems from significant application areas. The software is designed to compute a few (k) eigenvalues with user specified features such as those of largest real part or largest magnitude. Storage requirements are on the order of $n \cdot k$ locations. No auxiliary storage is required. A set of Schur basis vectors for the desired k-dimensional eigen-space is computed which is numerically orthogonal to working precision. Numerically accurate eigenvectors are available on request.

Important Features:

- Reverse Communication Interface.
- Single and Double Precision Real or Complex Arithmetic Versions for Hermitian, Non-Hermitian, Standard or Generalized Problems.
- Routines for Banded Matrices - Standard or Generalized Problems.
- Routines for The Singular Value Decomposition.
- Example driver routines that may be used as templates to implement numerous Shift-Invert strategies for all problem types, data types and precision.

A comparitive computational study of software available for large scale non-symmetric eigenvalue calculation was done by Lehoucq and Scott in [8]. They found ARPACK to be highly competitive

with all available software. In some cases it outperformed the other codes by orders of magnitude (fewer matrix-vector operations and/or cpu time).

There is also an efficient parallel version of the ARPACK called P_ARPACK. This parallel implementation is portable across a wide range of distributed memory platforms and only involves minimal changes to the serial code. The communication layers used for message passing are the Basic Linear Algebra Communication Subprograms (BLACS) developed for the ScaLAPACK project and Message Passing Interface(MPI). The parallelization paradigm found to be most effective for ARPACK on distributed memory machines was to provide the user with a Single Program Multiple Data (SPMD) template. The CGS orthogonalization process is parallelized by distributing the rows of $\mathbf{V}$ in a straightforward way across processors and using a distributed parallel DGEMV operation. The projected matrix $\mathbf{H}$ and associated operations are replicated on each processor. The reverse communication interface is one of the most important aspects in the design of ARPACK and this feature lends itself to a simplified SPMD parallelization strategy. It is relatively straightforward for the the user adapt the parellel template to a particular application. The reverse communication interface feature of ARPACK allows the P_ARPACK codes to be parallelized internally without imposing a fixed parallel decomposition on the matrix or the user supplied matrix-vector product. Memory and communication management for the matrix-vector product can be optimized independent of P_ARPACK . This feature enables the use of various matrix storage formats as well as calculation of the matrix elements on the fly.

The calling sequence to ARPACK remains unchanged except for the addition of the BLACS context (or MPI communicator). Inclusion of the context (or communicator) is necessary for global communication as well as managing I/O. The addition of the context is new to this implementation and reflects the improvements and standardizations being made in message passing [6, 2]. The parallel code has been tested on IBM SP2, CRAY T3D, SGI Cluster, Intel Paragon. For details see [11].

5 SPECTRAL TRANSFORMATIONS

Implicit restarting can be very effective at computing extremal eigenvalues. However, convergence is often slow and the method can even fail to converge when eigenvalues are clustered or when interior eigenvalues are sought. Convergence may be enhanced dramatically through a spectral transformation. For a generalized problem, this is required. If our problem is to find the generalized eigenvalues of $\mathbf{Ax} = \mathbf{Mx}\lambda$ then it is far more effective to find eigenpairs $(\mathbf{x}, \nu)$ of the shift invert operator

$$\mathbf{S} = (\mathbf{A} - \sigma\mathbf{M})^{-1}\mathbf{M}$$

and then the pairs $(\mathbf{x}, \lambda)$ where $\lambda = \sigma + \frac{1}{\nu}$ will be generalized eigenpairs for the original problem. Convergence is usually very rapid towards eigenvalues ν of $\mathbf{S}$ that are of largest magnitude and these transform to eigenvalues λ that are nearest the selected point σ. The details and nuances of this technique are far beyond the scope of this paper but they are discussed in [18]. $\mathbf{M} = \mathbf{I}$ in the remainder of this discussion.

Current research is involved with improved use and implementation of the spectral transformation. One the most pressing problems in the area of large scale eigenvalue computations is to devise a method to get effect of a rational spectral transformation without solving shift-invert equations accurately. Three methods are discussed here: The Rational Krylov, Jacobi-Davidson, and truncated RQ methods.

One of the most effective and promising methods for making better use of the spectral transformation is the Rational Krylov Method (RKS) introduced by Ruhe [13], [14]. The basic recursion

involved in RKS may be characterized by the equation [13]

$$\mathbf{A}\mathbf{V}_{k+1}\hat{\mathbf{H}}_k = \mathbf{V}_{k+1}\hat{\mathbf{G}}_k,$$

where $\mathbf{V}_{k+1}$ is n by $k+1$, $\hat{\mathbf{H}}_k$ and $\hat{\mathbf{G}}_k$ are $k+1$ by k, Hessenberg matrices and $\mathbf{V}_{k+1}^H\mathbf{V}_{k+1} = \mathbf{I}_{k+1}$. This relation is produced by a sequence of Arnoldi-like steps but the subspace that is generated is not a Krylov space. Instead the projective space is the linear span of a sequence of rational functions of $\mathbf{A}$ applied to the starting vector $\mathbf{v}_1$. The additional freedom from having two projected matrices $\mathbf{H}$ and $\mathbf{G}$ allows multiple shifts (spectral transformations) to be incorporated into one k-step RKS factorization. Only one shift can be used in conjunction with the Arnoldi method. Linear equations must be solved at each step of the iteration and refactorizatons of shifted operators are done every so often as new shift points are selected. The RKS method constructs an orthonormal basis for the Rational Krylov Subspace

$$\mathcal{S}(\mathbf{A}, \mathbf{v}_0, \{\mu_j\}) = \text{span}\{\mathbf{v}_0, \mathbf{v}_1, \mathbf{v}_2, \ldots \mathbf{v}_k\},$$

where $\mathbf{v}_j = (\mathbf{A} - \mu_j\mathbf{I})^{-1}\mathbf{v}_{j-1}$. The generalized eigen-pairs of $(\mathbf{H}_k, \mathbf{G}_k)$ (the leading $k \times k$ submatrices of the projections) provide Ritz approximations from this subspace.

Another new method that is very promising with respect to avoiding the need for accurate solutions of the shift-invert equations is the Jacobi-Davidson method [16]. This method is a generalization of the Davidson method that is quite popular within computational chemistry. Given an initial approximation $\mathbf{v}_0$ of a desired eigenvector, the Jacobi-Davidson method finds, at each step, a correction vector $\mathbf{z}_k$ that is orthogonal to the previous approximate eigenvector $\mathbf{u}_k$. This vector is added into a subspace from which the next approximate eigenpairs are drawn. The correction vector is solved from the equation

$$(\mathbf{I} - \mathbf{u}_k\mathbf{u}_k^H)(\mathbf{A} - \lambda\mathbf{I})(\mathbf{I} - \mathbf{u}_k\mathbf{u}_k^H)\mathbf{z}_k = -\mathbf{r}_k \quad \text{and} \quad \mathbf{z}_k \perp \mathbf{u}_k, \tag{7}$$

where $\mathbf{r}_k = \mathbf{A}\mathbf{u}_k - \theta_k\mathbf{u}_k$, and θ_k is the current approximation to the eigenvalue of interest. It can be shown [16] that if (7) is solved exactly, the Jacobi-Davidson method becomes equivalent to RKS with an optimal shift selected in each iteration [14]. The advantages of this approach are that the correction equation does not need to be solved to any particular accuracy and that the projected equations are likely to be better conditioned than the unprojected ones because the "singularity" is being projected out. However, if the equations are not solved accurately, the Krylov relation is completely lost and the Hessenberg relationships (1) are not preserved with the Jacobi-Davidson iteration. To obtain several eigenvalues and eigenvectors, some standard deflation schemes [15] are needed. To avoid building a large dimensional subspace $\mathcal{S}$, restarting is also necessary. The implementation of the Jacobi Davidson QR (JDQR) algorithm is explained in detail in [5].

The final method discussed here is the "truncated RQ-iteration" [19]. This new method is in the spirit of implicit restarting but recasts the underlying iteration so that rapid inverse-iteration-like convergence occurs in the leading columns of the updated Arnoldi iteration. This scheme encompasses important features from the Rational Krylov scheme when equations can be solved accurately. It also shares many of the features of the Jacobi-Davidson method when inexact solutions to shift-invert equations are introduced.

The TRQ equations may be used to develop a truncated k-step version of the Implicitly Shifted RQ-iteration. If a k-step Arnoldi factorization (1) has been obtained then a k-step TRQ iteration may be implemented as shown in Algorithm 3 (Figure 3.)

Implicit restarting a truncation of the (implicitly) shifted QR-iteration. There is a related iteration, the shifted RQ-iteration, which as the name suggest is QR in reverse. After an initial reduction of $\mathbf{A}$ to Hessenberg form $\mathbf{AV} = \mathbf{VH}$, it amounts to

$$\mathbf{H} - \mu_j\mathbf{I} = \mathbf{RQ}; \quad \mathbf{H} \leftarrow \mathbf{QR} + \mu_j\mathbf{I}; \quad \mathbf{V} \leftarrow \mathbf{VQ}^H; \quad \text{for} \quad j = 1, 2, \cdots \quad \text{until convergence.}$$

Algorithm 3: (TRQ) Truncated RQ-iteration

Input: $(\mathbf{A}, \mathbf{V}_k, \mathbf{H}_k, \mathbf{f}_k)$ with $\mathbf{AV}_k = \mathbf{V}_k\mathbf{H}_k + \mathbf{f}_k\mathbf{e}_k^T$, $\mathbf{V}_k^H\mathbf{V}_k = \mathbf{I}$, $\mathbf{H}_k$ upper Hessenberg.
Output: $(\mathbf{V}_k, \mathbf{H}_k)$ such that $\mathbf{AV}_k = \mathbf{V}_k\mathbf{H}_k$, $\mathbf{V}_k^H\mathbf{V}_k = \mathbf{I}$ and $\mathbf{H}_k$ is upper triangular.

1. Put $\beta_k = \|\mathbf{f}_k\|$ and put $\mathbf{v} = \mathbf{f}_k/\beta_k$;
2. **for** $j = 1, 2, 3, \ldots$ until *convergence*,
   2.1. Select a shift $\mu \leftarrow \mu_j$;
   2.2. Solve $\begin{pmatrix} \mathbf{A}-\mu\mathbf{I} & \mathbf{V}_k \\ \mathbf{V}_k^H & 0 \end{pmatrix}\begin{pmatrix} \mathbf{v}_+ \\ -\mathbf{h} \end{pmatrix} = \begin{pmatrix} \mathbf{v}\alpha \\ 0 \end{pmatrix}$ with $\|\mathbf{v}_+\| = 1$;
   2.4. *RQ* Factor $\begin{pmatrix} \mathbf{H}_k - \mu\mathbf{I}_k & \mathbf{h} \\ \beta_k e_k^T & \alpha \end{pmatrix} = \begin{pmatrix} \mathbf{R}_k & \mathbf{r} \\ 0 & \rho \end{pmatrix}\begin{pmatrix} \mathbf{Q}_k & \mathbf{q} \\ \sigma\mathbf{e}_k^T & \gamma \end{pmatrix}$;
   2.5. $\mathbf{V}_k \leftarrow \mathbf{V}_k\mathbf{Q}_k^H + \mathbf{v}_+\mathbf{q}^H$;
   2.6. $\beta_k \leftarrow \sigma\mathbf{e}_k^T\mathbf{R}_k\mathbf{e}_k$; $\mathbf{v} \leftarrow \mathbf{v}_k\bar{\sigma} + \mathbf{v}_+\bar{\gamma}$;
   2.7. $\mathbf{H}_k \leftarrow \mathbf{Q}_k\mathbf{R}_k + \mu\mathbf{I}_k$;
3. **end**;

Figure 3: The Truncated RQ-iteration.

As with QR, the matrix $\mathbf{H}$ remains Hessenberg througout the RQ-iteration. The primary difference is that

$$\mathbf{v}_1^+ \leftarrow \frac{1}{\rho_{11}}(\mathbf{A} - \mu_j\mathbf{I})^{-1}\mathbf{v}_1$$

so there is an inverse iteration relation amongst the successive first columns of $\mathbf{V}$. If this iteration can be truncated in the a manner analogous to the truncation of QR-iteration in implicit restarting then a rapidly convergent technique for computing several eigenvalues and vectors would result.

The key idea here is to determine the $k+1$-st column $\mathbf{v}_+$ of the updated matrix $\mathbf{V}$ and the $k+1$-st column of $\mathbf{H}$ that would have been produced in the RQ-iteration if carried out in full. In [19] the necessary equations were developed and analyzed. These *TRQ equations* are used to develop a truncated k-step version of the Implicitly Shifted RQ-iteration. If a k-step Arnoldi factorization (1) has been obtained then a k-step TRQ iteration may be implemented as shown in Algorithm 3 (Figure 3.)

At each iteration, the TRQ equations are solved and then the iteration is completed through the normal RQ bulge chase sweep. As eigenvalues converge, the standard deflation rules of the RQ-iteration may be applied. Orthogonality of the basis vectors is explicitly maintained through accurate solution of the defining equation. Moreover, even if the accuracy of this solution is relaxed, orthogonality may be enforced explicitly through the DGKS mechanism [1]. Potentially, the linear solve indicated at Step 2.2 of Algorithm 3 could be provided by a straightforward block elimination scheme. However, considerable refinements to this scheme are possible due to the existing k-step Arnoldi relationship (1).

The TRQ equations may be solved very efficiently as shown in Algorithm 4.

This approach is appropriate when sparse direct factorization of $(\mathbf{A} - \mu\mathbf{I})$ is feasible. When this is not the case we must resort to an iterative scheme. For an iterative scheme, there may be an advantage to solving the projected equation

$$(I - V_kV_k^H)(A - \mu I)(I - V_kV_k^H)\hat{w} = f_k$$

Algorithm 4: Direct Solution of the TRQ Equations

Input: $(\mathbf{A}, \mathbf{V}_k, \mathbf{H}_k, \mathbf{f}_k, \mu)$ with $\mathbf{AV}_k = \mathbf{V}_k\mathbf{H}_k + \mathbf{f}_k e_k^T$, $\mathbf{V}_k^H\mathbf{V}_k = \mathbf{I}$ and $\mathbf{V}_k^H\mathbf{f}_k = 0$.
Output: $(\mathbf{v}_+, \mathbf{h}, \alpha)$ such that $(\mathbf{A} - \mu\mathbf{I})\mathbf{v}_+ = \mathbf{V}_k h +_{\mathbf{k}} \alpha$, $\mathbf{V}_k^H\mathbf{v}_+ = 0$ and $\|\mathbf{v}_+\| = 1$.

1. Choose $\mathbf{t}$ and η and solve $(\mathbf{A} - \mu\mathbf{I})\mathbf{w} = \mathbf{V}_k\mathbf{t} + \mathbf{f}_k\eta$;
2. $\mathbf{y} \leftarrow \mathbf{V}_k^H\mathbf{w}$;
3. $\mathbf{w} \leftarrow \mathbf{w} - \mathbf{V}_k\mathbf{y}$;
4. $\mathbf{v}_+ \leftarrow \frac{\mathbf{w}}{\|\mathbf{w}\|}$; $\alpha \leftarrow \mathbf{f}_k^H(\mathbf{A} - \mu\mathbf{I})v_+/\|\mathbf{f}_k\|$; $\mathbf{h} \leftarrow \mathbf{V}_k^H\mathbf{A}\mathbf{v}_+$;

Figure 4: Direct Solution of the TRQ Equations.

and putting

$$v_+ \leftarrow \frac{w}{\|w\|};$$

where $w = (I - V_kV_k^H)\hat{w}$. This is mathematically equivalent to solving the TRQ equations. As with the Jacobi-Davidson scheme, the advantage here is that the matrix

$$(I - V_kV_k^H)(A - \mu I)(I - V_kV_k^H)$$

is most likely to be much better conditioned than $(A - \mu I)$ when μ is near an eigenvalue of A. In [19] a restart mechanism is developed for method that allows this TRQ equation to be solved inaccurately (with a pre-conditioned iterative method say) and the results are quite competitive with the JDQR approach.

6 CONCLUSIONS

The past few years have seen substantial progress. It is now possible to solve problems on the order of $n = 10^6$. In fact ARPACK with Tchebycheff polynomial preconditioning was recently used to compute the three lowest non-trivial eigenvectors of a weighted Laplacian matrix of a graph. The matrix was of dimension 2.4 million, with about 44 million nonzeros. The calculation took just under 44 hours on an SGI Onyx with two R10000 processors and 3Gbytes of RAM. The results were needed as part of a project to visually represent the large-scale structure of scientific disciplines based upon citation analysis [7].

The nonsymmetric problem is in far better shape than it was just five years ago. However, there is still considerable research to be done before there will be sufficient tools available to solve "any" problem. Clearly, polynomial pre-conditioning, inexact spectral transformations, and other acceleration schemes need to be developed and improved to handle the most difficult problems.

References

[1] J. Daniel, W. B. Gragg, L. Kaufman, and G. W. Stewart. Reorthogonalization and stable algorithms for updating the Gram–Schmidt QR factorization. *Mathematics of Computation*, 30:772–795, 1976.

[2] J. J. Dongarra and R. C. Whaley. Lapack working note 94, a user's guide to the blacs v1.0. Technical report, Dept. Computer Science., University of Tennessee, Knoxville, TN, 1995.

[3] J.J. Dongarra, J. DuCroz, S. Hammarling, and R. J. Hanson. An extended set of Fortran basic linear algebra subprograms. *ACM Transactions on Mathematical Software*, 14(1):1–17, 1988.

[4] J.J. Dongarra, I. S. Duff, D. C. Sorensen, and H. A. Van der Vorst. *Solving Linear systems on Vector and shared memory computers.* SIAM, Philadelphia, PA., 1991.

[5] D. R. Fokkema, G. L. G. Sleijpen, and H.A. Van der Vorst. A Jacobi-Davidson style QR and QZ algorithm for partial reduction of matrix pencils. Technical Report 941, University Utrecht, Department of Mathematics, 1996.

[6] Message Passing Interface Forum. Mpi: A message-passing interface standard. *International Journal of Supercomputer Applications and High Performance Computing*, 8, 1994.

[7] B. Hendrickson. private communication, Sandia National Laboratory, Albuquerque, NM, 1997.

[8] R. B. Lehoucq and J. A. Scott. An evaluation of software for computing eigenvalues of sparse nonsymmetric matrices. Preprint MCS-P547-1195, Argonne National Laboratory, Argonne, IL, 1996.

[9] R. B. Lehoucq and D. C. Sorensen. Deflation techniques within an implicitly restarted iteration. *SIAM Journal on Matrix Analysis and Applications*, 17(4):789–821, 1996.

[10] R. B. Lehoucq, D. C. Sorensen, P. Vu, and C. Yang. ARPACK: *An implementation of the Implicitly Re-started Arnoldi Iteration that computes some of the eigenvalues and eigenvectors of a large sparse matrix*, 1995. Available from ftp.caam.rice.edu under the directory pub/software/ARPACK.

[11] K. Maschhoff and D. C. Sorensen. P_ARPACK: An efficient portable large scale eigenvalue package for distributed memory parallel architectures. In J. Wasniewski, K. Madsen J. Dongarra, and D. Olesen, editors, *Applied Parallel Computing in Industrial Problems and Optimization*, volume 1184, pages 478–496, Berlin, 1996. Springer–Verlag.

[12] B. N. Parlett. *The Symmetric Eigenvalue Problem.* Prentice-Hall, 1980.

[13] A. Ruhe. Rational Krylov sequence methods for eigenvalue computations, II: Matrix pairs. *Linear Algebra and Its Applications*, 197,198:283–295, 1994.

[14] A. Ruhe. Rational Krylov, a practical algorithm for large sparse nonsymmetric matrix pencils. Technical Report UCB/CSD-95-871 (revised), Computer Science Division(EECS), University of California Berkeley, CA 94720, 1995.

[15] Y. Saad. *Numerical Methods for Large Eigenvalue Problems.* Halsted Press, 1992.

[16] G. L. G. Sleijpen and H.A. Van der Vorst. A Jacobi-Davidson iteration method for linear eigenvalue problems. *SIAM Journal on Matrix Analysis and Applications*, 17(2), April 1996.

[17] D. C. Sorensen. Implicit application of polynomial filters in a k-step Arnoldi method. *SIAM Journal on Matrix Analysis and Applications*, 13(1):357–385, January 1992.

[18] D. C. Sorensen. Implicitly restarted Arnoldi/Lanczos methods for large scale eigenvalue calculations. In D. E. Keyes, A. Sameh, and V. Venkatakrishnan, editors, *Proceedings of the ICASE/LaRC Workshop on Parallel Numerical Algorithms, May 23-25, 1994*, Norfolk, Va, 1995. Kluwer. To appear.

[19] D. C. Sorensen and C. Yang. A truncated RQ-iteration for large scale eigenvalue calculations. Technical Report TR96-06, Dept. Computational and Applied Math., Rice University, Houston, TX 77251, 1996.