IMPLEMENTATION OF AN IMPLICITLY RESTARTED BLOCK ARNOLDI METHOD*

R. B. LEHOUCQ† AND K. J. MASCHHOFF‡

Abstract. Although block Arnoldi methods and their symmetric Lanczos counterparts are not new, studies comparing their relative strengths and merits are lacking. In this report, we investigate the effect of changing the block size when computing a selected portion of the eigenvalues (and associated invariant subspace) of a large sparse matrix within the framework of implicit restarting. In particular, we demonstrate how Sorensen's implicitly restarted Arnoldi method may be extended to block formulations. Our experimental results indicate that our implicitly restarted scheme is superior to other block methods.

Key words. Arnoldi method, Lanczos method, block methods, eigenvalues, deflation, implicit restarting

AMS subject classifications. 65F15, 65G05

1. Introduction.

This report considers the use of both blocked and unblocked Arnoldi reductions for large-scale eigenvalue computations. Block methods are used for two major reasons. The first one is for reliably determining multiple and/or clustered eigenvalues. The second reason is related to issues dealing with computational efficiency. In many instances, the cost of computing a few matrix-vector products is commensurate with that of one matrix-vector product. On the other hand, two major drawbacks of block methods are the (not insignificant) added computational complexity of the software implementation and the comparative lack of theoretical understanding. There also remains the selection of the block size.

Recent work [15] indicates that an unblocked Arnoldi method coupled with a deflation strategy may be used reliably to compute multiple and/or clustered eigenvalues. However, an unblocked reduction may prove inefficient for some eigenvalue problems because of the cost of computing the Arnoldi subspace. Moreover, a relatively small convergence tolerance (no larger than the square root of machine precision) is required to reliably compute nearby eigenvalues. Many problems do not require this much accuracy, and such a criterion can result in unnecessary computation.

For symmetric matrices, Cullum and Donath [4] propose what appears to be the first block Lanczos method. Golub and Underwood [11] also discuss the method. The first industrial implementation is in [12].

The history for nonsymmetric matrices is much shorter. Sadkane [25] discusses a block Arnoldi algorithm within a restarting method. The books by Chatelin [3] and Saad [24] also include material for block Arnoldi/Lanczos methods. More recently, Scott [27] provides software for a block Arnoldi method.

An independent but parallel development of our ideas is laid out in the report by Baglama, Calvetti, Reichel, and Ruttan [2]. However, their study considers only the

* The work of R. B. Lehoucq was supported by the Mathematical, Information, and Computational Sciences Division subprogram of the Office of Computational and Technology Research, U.S. Department of Energy, under Contract W-31-109-Eng-38. The work of K. J. Maschhoff was supported in part by NSF cooperative agreement CCR-9120008, NSF contract ASC-9408795, and by ARPA contract number DAAL03-91-C-0047 (administered by the U.S. Army Research Office).

† Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL 60439 lehoucq@mcs.anl.gov, http://www.mcs.anl.gov/home/lehoucq/index.html.

‡ Computational and Applied Mathematics Department, Rice University, Houston, TX 77251 kristyn@caam.rice.edu, http://www.caam.rice.edu/kristyn.

symmetric case, while our research considers nonsymmetric matrices.

The article is organized as follows. We first consider the careful numerical implementation of block methods. Robust deflation, orthogonalization, and restarting strategies are presented. In particular, we demonstrate how Sorensen's implicitly restarted Arnoldi method [29] may be extended to a block one.

Finally, we perform a series of numerical experiments to assess the differences between the blocked and unblocked variants. The goal of our study is to provide the numerical analyst and software developer a better understanding of the many issues involved.

2. Notation and Definitions. This section establishes the basic notation to be used in this article. We employ Householder notational conventions. Capital and lower-case letters denote matrices and vectors, respectively, while lower-case Greek letters denote scalars. All matrices consist of complex numbers unless stated otherwise.

The order of $\mathbf{A}$ will always be denoted by n. The identity matrix of order m is denoted by $\mathbf{I}_m$. The jth canonical basis vector is denoted by $\mathbf{e}_j$, the jth column of the identity matrix, and $\mathbf{E}_j \equiv \begin{bmatrix} \mathbf{e}_{(j-1)b+1} & \cdots & \mathbf{e}_{jb} \end{bmatrix}$, where b is a positive integer. We will call the block size b.

A matrix of lower bandwidth b will be called a banded upper Hessenberg matrix. We drop "upper" when the context is clear. Omission of the word *band* implies that the block size is one. We say that a band Hessenberg matrix is unreduced if all the elements on the bth subdiagonal are nonzero.

We now define several matrices that will prove useful. $\mathbf{H}_j$ denotes a band Hessenberg matrix of order bj of lower bandwidth b; $\mathbf{T}_j$ denotes an upper triangular matrix of order j, regardless of any block size b; and $\mathbf{F}_j$ and $\mathbf{U}_j$ denote matrices with n rows and b columns, where the subscript acts as an index. On the other hand, $\mathbf{V}_j$ denotes a matrix with n rows and bj columns. $\mathbf{U}_j$ denotes the jth block of b vectors of $\mathbf{V}_m$, and $\mathbf{G}_{i,j}$ denotes the square matrix of order b located in the i,jth block of order b of $\mathbf{H}_m$. Note that $\mathbf{G}_{j+1,j}$ is an upper triangular matrix. These matrices will define the dimensions of other matrices used in this article.

The transpose of a vector $\mathbf{x}$ is denoted by $\mathbf{x}^T$, and the complex conjugate of $\mathbf{x}^T$ is denoted by $\mathbf{x}^H$. The norm used is the Euclidean one denoted by $\|\cdot\|$.

3. Block Arnoldi Reductions. Let $\mathbf{A}$ be a matrix of order n and $b > 0$ be the block size. We say that

$$\mathbf{A}\mathbf{V}_m = \mathbf{V}_m\mathbf{H}_m + \mathbf{F}_m\mathbf{E}_m^T \tag{3.1}$$

is a block Arnoldi reduction of length m when $\mathbf{V}_m^H\mathbf{A}\mathbf{V}_m = \mathbf{H}_m$ is a banded upper Hessenberg matrix, $\mathbf{V}_m^H\mathbf{V}_m = \mathbf{I}_{mb}$, and $\mathbf{V}_m^H\mathbf{F}_m = \mathbf{0}$. Let $\mathbf{U}_{m+1}\mathbf{G}_{m+1,m}$ denote the QR factorization of $\mathbf{F}_m$. Using the notation established in § 2, we have

$$\mathbf{A}\,\mathbf{V}_m = \begin{bmatrix} \mathbf{U}_1 & \cdots & \mathbf{U}_m \end{bmatrix} \begin{bmatrix} \mathbf{G}_{1,1} & \cdots & \cdots & \mathbf{G}_{1,m} \\ \mathbf{G}_{2,1} & \ddots & \vdots & \vdots \\ \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \cdots & \mathbf{G}_{m,m-1} & \mathbf{G}_{m,m} \end{bmatrix}$$
$$+\mathbf{U}_{m+1}\mathbf{G}_{m+1,m}\mathbf{E}_m^T.$$

- Let $\mathbf{AV}_m = \mathbf{V}_m\mathbf{H}_m + \mathbf{F}_m\mathbf{E}_m^T$ be a length-m block Arnoldi reduction where $\mathbf{V}_m^H\mathbf{F}_m = \mathbf{0}$.
  1. Compute the QR factorization $\mathbf{U}_{m+1}\mathbf{G}_{m+1,m} = \mathbf{F}_m$ using iterated classical Gram-Schmidt.
  2. $\mathbf{V}_{m+1} = \begin{bmatrix} \mathbf{V}_m & \mathbf{U}_{m+1} \end{bmatrix}$.
  3. $\mathbf{W} = \mathbf{AU}_{m+1}$ and $\mathbf{G}_{m+1,m+1} = \mathbf{U}_{m+1}^H\mathbf{W}$.
  4. $\mathbf{H}_{m+1} = \begin{bmatrix} \mathbf{H}_m & \mathbf{V}_m^H\mathbf{W} \\ \mathbf{G}_{m+1,m}\mathbf{E}_m^T & \mathbf{G}_{m+1,m+1} \end{bmatrix}$.
  5. $\mathbf{F}_{m+1} = \mathbf{W} - \mathbf{V}_{m+1}\begin{bmatrix} \mathbf{V}_m^H\mathbf{W} \\ \mathbf{G}_{m+1,m+1} \end{bmatrix}$.

FIG. 3.1. *Extending a Block Arnoldi Reduction*

The columns of $\mathbf{V}_m$ are an orthogonal basis for the block Krylov subspace

$$\mathcal{K}_m(\mathbf{A}, \mathbf{U}_1) \equiv \{\mathbf{U}_1, \mathbf{AU}_1, \cdots, \mathbf{A}^{m-1}\mathbf{U}_1\}.$$

If $m > \bar{m} \equiv \text{ceiling}(n/b)$, then $\mathbf{F}_m = \mathbf{0}$ and $\mathbf{H}_{\bar{m}}$ is the orthogonal reduction of $\mathbf{A}$ into banded upper Hessenberg form. We assume, for the moment, that $\mathbf{F}_m$ is of full rank and further suppose that the diagonal elements of $\mathbf{G}_{m+1,m}$ are positive. Thus, a straightforward extension of the implicit Q theorem [9, pp. 367–368] gives that $\mathbf{F}_m$ is (uniquely) specified by the starting block $\mathbf{U}_1$. Note that if $\mathbf{A} = \mathbf{A}^H$, then $\mathbf{H}_m$ is a block tridiagonal matrix. Figure 3.1 lists an algorithm to compute a block Arnoldi reduction.

The QR factorization in Step 1 is computed via an iterated classical Gram-Schmidt (CGS) algorithm using a possible correction step. See [5] for details and the simple test used to determine whether a correction step is necessary. One benefit of this scheme is that it allows the use of the Level 2 BLAS [7] matrix-vector multiplication subroutine `_GEMV`. Moreover, this scheme also gives a simple way to fill out a rank-deficient $\mathbf{F}_m$. If a third step of orthogonalization is needed when generating column j of $\mathbf{U}_{m+1}$, then the corresponding column of $\mathbf{F}_m$ is linearly dependent on the previous $j-1$ columns of $\mathbf{U}_{m+1}$. The jth diagonal element of $\mathbf{G}_{m+1,m}$ is set to zero, and a random unit vector is orthogonalized against $\mathbf{V}_m$ and the first $j-1$ columns of $\mathbf{U}_{m+1}$.

Step 3 allows the application of $\mathbf{A}$ to a group of vectors. This might prove essential when accessing $\mathbf{A}$ is expensive. Clearly, the goal is to amortize the cost of applying $\mathbf{A}$ over several vectors.

As written, Step 5 is one step of block classical Gram-Schmidt (bCGR). This allows the use of the Level 3 BLAS [6] matrix-matrix multiplication subroutine `_GEMM` for computing $\mathbf{V}_m^H\mathbf{W}$. To ensure the orthogonality of $\mathbf{F}_{m+1}$ with $\mathbf{V}_{m+1}$, a second step of bCGR is performed except when $b = 1$. In this latter case, the simple test in DGKS [5] is used to determine whether a second orthogonalization step is needed. See [16] for details.

The scheme given for computing $\mathbf{V}_{m+1}$ is equivalent to the one proposed by Ruhe [20] and is the one used by the implicitly restarted block Lanczos code [2]. Although the approach in [2] cleanly deals with the problem of rank deficient $\mathbf{F}_m$, the implementation does not exploit the ability to apply $\mathbf{A}$ as in Step 3 above. Instead, as proposed in [20], $\mathbf{A}$ is applied to each column of $\mathbf{F}_m$ followed by computing the corresponding column of $\mathbf{U}_{m+1}$ and $\mathbf{H}_{m+1}$. Our implementation reorganizes Ruhe's approach so that the computation of the matrix of coefficients $\mathbf{V}_m^H \mathbf{W}$ is separated from the QR factorization of $\mathbf{F}_m$. The advantage is that Steps 3–5 reduce the cost of I/O by a factor of the blocksize and increase the amount of floating-point operations per memory reference.

4. Restarting a Block Arnoldi Reduction. For symmetric $\mathbf{A}$, Saad [21] shows that as m increases, the quality of the extremal eigenvalues of $\mathbf{H}_m$ improves to the well-separated extremal eigenvalues of $\mathbf{A}$. For the unblocked Arnoldi reduction, he also shows [23] a similar improvement for increasing m. Unfortunately, given a large value of n, the value of m needed for a desired degree of approximation may be impractical because of storage constraints required for the Arnoldi/Lanczos vectors. This situation is particularly exacerbated for nonsymmetric $\mathbf{A}$. A way to alleviate the cost of building a large reduction is to periodically *restart* the reduction.

Although restarting an Arnoldi/Lanczos reduction is not a new idea, only recently with the availability of ARPACK [16] has it become a powerful method for computing selected eigenvalues (and eigenvectors) of a large sparse matrix. Karush [13] proposes what appears to be the first example of a restarted iteration for use with the Lanczos reduction. Saad [22] investigated its use for nonsymmetric matrices. Repeatedly restarting a reduction gives an iteration defined by a two-stage process. First, a block Arnoldi/Lanczos reduction of length $m < n$ is computed. From the information available in this reduction, a subsequent reduction is computed. This defines the iteration and is deemed successful if improved estimates to the eigenvalues of $\mathbf{A}$ appear in the subsequent reductions.

Sorensen [29] proposed restarting a reduction in an implicit fashion. A high-quality software implementation is found in the software package ARPACK [16]. We now develop an implicitly restarted block Arnoldi iteration, extending the results in [29].

Suppose that μ is a real shift, and let $\mathbf{H}_{m+1} - \mu \mathbf{I} = \mathbf{QR}$ with $\mathbf{Q}$ orthogonal and $\mathbf{R}$ upper triangular matrices, respectively. Then from (3.1)

$$(\mathbf{A} - \mu\mathbf{I})\mathbf{V}_{m+1} - \mathbf{V}_{m+1}(\mathbf{H}_{m+1} - \mu\mathbf{I}) = \mathbf{F}_{m+1}\mathbf{E}_{m+1}^T, \tag{4.1}$$

$$(\mathbf{A} - \mu\mathbf{I})\mathbf{V}_{m+1} - \mathbf{V}_{m+1}\mathbf{QR} = \mathbf{F}_{m+1}\mathbf{E}_{m+1}^T, \tag{4.2}$$

$$(\mathbf{A} - \mu\mathbf{I})(\mathbf{V}_{m+1}\mathbf{Q}) - (\mathbf{V}_{m+1}\mathbf{Q})(\mathbf{RQ}) = \mathbf{F}_{m+1}\mathbf{E}_{m+1}^T\mathbf{Q},$$

$$\mathbf{A}(\mathbf{V}_{m+1}\mathbf{Q}) - (\mathbf{V}_{m+1}\mathbf{Q})(\mathbf{RQ} + \mu\mathbf{I}) = \mathbf{F}_{m+1}\mathbf{E}_{m+1}^T\mathbf{Q}. \tag{4.3}$$

The matrices are updated via $\mathbf{V}_{m+1}^+ \leftarrow \mathbf{V}_{m+1}\mathbf{Q}$ and $\mathbf{H}_{m+1}^+ \leftarrow \mathbf{RQ} + \mu\mathbf{I}$, and the latter matrix remains upper band Hessenberg, since $\mathbf{R}$ is upper triangular and $\mathbf{Q}$ is upper band Hessenberg. Partitioning the matrices in the updated equation results in

$$\mathbf{A}\begin{bmatrix} \mathbf{V}_m^+ & \mathbf{V}_{m+1}^+\mathbf{E}_{m+1} \end{bmatrix} = \begin{bmatrix} \mathbf{V}_m^+ & \mathbf{V}_{m+1}^+\mathbf{E}_{m+1} \end{bmatrix} \begin{bmatrix} \mathbf{H}_m^+ & \mathbf{C} \\ \mathbf{R}_{m+1}^+\mathbf{E}_m^T & \mathbf{B} \end{bmatrix} + \mathbf{F}_{m+1}\begin{bmatrix} \mathbf{0} & \cdots & \mathbf{E}_{m+1}^T\mathbf{Q}\mathbf{E}_m & \mathbf{E}_{m+1}^T\mathbf{Q}\mathbf{E}_{m+1} \end{bmatrix}, \tag{4.4}$$

revealing that the last $2 \cdot b$ columns of the rightmost matrix are no longer zero ones. However, if we equate the first $m \cdot b$ columns of (4.4), it follows that

$$\mathbf{A}\mathbf{V}_m^+ = \mathbf{V}_m^+\mathbf{H}_m^+ + (\mathbf{V}_{m+1}^+\mathbf{E}_{m+1}\mathbf{R}_{m+1}^+ + \mathbf{F}_{m+1}\mathbf{E}_{m+1}^T\mathbf{Q}\mathbf{E}_m)\mathbf{E}_m^T. \tag{4.5}$$

If we perform the update $\mathbf{F}_m^+ \leftarrow \mathbf{V}_{m+1}^+\mathbf{E}_{m+1}\mathbf{R}_{m+1}^+ + \mathbf{F}_{m+1}\mathbf{E}_{m+1}^T\mathbf{Q}\mathbf{E}_m$ and note that $(\mathbf{V}_m^+)^T\mathbf{F}_m^+ = \mathbf{0}$, it follows that Equation (4.5) is a length m block Arnoldi reduction.

- Input: A length r block Arnoldi reduction $\mathbf{AV}_r^{(1)} = \mathbf{V}_r^{(1)}\mathbf{H}_r^{(1)} + \mathbf{F}_r^{(1)}\mathbf{E}_r^T$.
- Iteration loop: For $i = 1, 2 \cdots$ until convergence
  1. Extend the length r block Arnoldi reduction by p blocks:
  $$\mathbf{AV}_{r+p}^{(i)} = \mathbf{V}_{r+p}^{(i)}\mathbf{H}_{r+p}^{(i)} + \mathbf{F}_{r+p}^{(i)}\mathbf{E}_{r+p}^T$$
  2. Compute the Schur decomposition of $\mathbf{H}_{r+p}^{(i)}$, and check for convergence of the k wanted eigenvalues.
  3. Lock the Ritz values that satisfy the convergence tolerance.
  4. Select p shifts, and determine r.
  5. Apply p implicit QR steps with shifts to $\mathbf{H}_{r+p}^{(i)}$. Retain a length r block Arnoldi reduction.

FIG. 5.1. *Implicitly Restarted Block Arnoldi Method (bIRAM)*

An induction argument shows that up to $p = m$ shifts may be applied in this manner, resulting in an updated length $m + 1 - p$ block Arnoldi reduction. A final $m + 1$-th shift may be applied without involving $\mathbf{A}$, since

$$(\mathbf{A} - \mu_{m+1}\mathbf{I})\mathbf{V}_1^+ = \mathbf{V}_1^+(\mathbf{H}_1^+ - \mu_{m+1}\mathbf{I}) + \mathbf{F}_1^+$$

and orthogonalizing the righthand side gives a new starting block. This approach was proposed in [1].

Denote by the polynomial $\Phi_p(\lambda) = (\lambda - \mu_1)\cdots(\lambda - \mu_p)$. Post-multiplication of (4.2) with $\mathbf{E}_1$ results in the important relationship

$$\Phi_p(\mathbf{A})\mathbf{V}_1 = \mathbf{V}_{m+1}\mathbf{QRE}_1 = \mathbf{V}_{m+1}\mathbf{Q}_1\mathbf{R}_1, \tag{4.6}$$

where $\mathbf{R}_1$ is the leading principal sub-matrix of $\mathbf{R}$ of order b and $\mathbf{Q}_1$ contains the first b columns of $\mathbf{Q}$. If the polynomial $\Phi_p(\lambda) = \lambda^p$ is used, the above relationship gives a connection with subspace iteration. Specifically, it shows how subspace iteration is implicitly performed on the current Arnoldi matrix $\mathbf{V}_{m+1}$. However, instead of restarting a subsequent Arnoldi reduction with $\mathbf{V}_{m+1}\mathbf{Q}_1$, we apply $\mathbf{Q}$ as in (4.3). Thus, a polynomial in $\mathbf{A}$ is applied implicitly—just as in the traditional implicitly shifted QR algorithm. The benefit is that matrix-vector products with $\mathbf{A}$ are avoided. If matrix-vector products with $\mathbf{A}$ are expensive, implicitly restarting leads to a more efficient algorithm. See [14] for details on a connection with subspace iteration and the QR algorithm.

5. Practical Algorithm.

Figure 5.1 lists a generic block implicitly restarted Arnoldi method (bIRAM). The remainder of this section discusses some of the implementation issues necessary for a robust software implementation. In particular, we address the issues of block size (b), the selection of shifts during the implicit restart and choice of p, convergence considerations, and a deflation strategy. The superscript (i) is dropped whenever convenient.

5.1. Block Size.

We now consider some of the issues and tradeoffs that should be considered when selecting the block size. For this discussion we assume that comparisons are made using a fixed maximum dimension for the subspace.

As the blocksize increases, the length of the Arnoldi reduction $m = r+p$ decreases. Since the degree of the largest power of $\mathbf{A}$ in the corresponding Krylov space is $m-1$, smaller block sizes allow a polynomials of larger degree to be applied. The downside to an unblocked method is that it cannot compute multiple copies of an eigenvalue of $\mathbf{A}$ unless the reduction already well approximates some of the associated eigenvectors. For example, the first Ritz pair should give a residual of $\mathcal{O}(\epsilon_M)$ or smaller relative to the norm of $\mathbf{A}$ before the second copy emerges.

One of the benefits of block methods is that they are more reliable for computing approximations to the clustered and/or multiple eigenvalues using a relatively large convergence criterion. Note that the block size used may be varied during each iteration.

5.2. Filter Choice. Each iteration of Algorithm (5.1) implicitly replaces the starting block $\mathbf{U}_1$ with $\Phi_p(\mathbf{A})\mathbf{U}_1$. Recall that the polynomial $\Phi_p(\mathbf{A})$ acts like a filter on the starting block amplifying components in the direction of the desired invariant subspace.

Numerous choices are possible for the selection of the p shifts, including the specific choice of p. If the shifts are in complex conjugate pairs, the implicit double shift [10, pp. 355–358] can be used to avoid complex arithmetic.

Typically, the p shifts are selected by utilizing the spectral information contained in $\mathbf{H}_{r+p}$. Partition the eigenvalues of $\mathbf{H}_m$ so that

$$\{\underbrace{\theta_1, \ldots, \theta_r}_{\text{wanted}}\} \cup \{\underbrace{\theta_{r+1}, \ldots, \theta_m}_{\text{unwanted}}\}. \tag{5.1}$$

For an unblocked reduction, the p shifts are selected from the unwanted eigenvalues of $\mathbf{H}_m$ where $r = k$. Sorensen [29] proposed this as a *exact shift* strategy. This strategy is equivalent to restarting the subsequent reduction with a linear combination of the approximate Schur vectors associated with the k wanted eigenvalues. Other choices of shifts are possible. These include the roots of a Chebyshev polynomial, harmonic Ritz values [17, 18, 28], and Leja points [1].

In the algorithm listed in Fig. 5.1, the integer r is typically set to k, the number of wanted eigenvalues, during the input step. Once the iteration loop has been entered, the values of r, p and thus $m = r + p$ may vary for every value of i. When $b > 1$, we cannot apply all $p = m \cdot b - k$ unwanted eigenvalues as shifts. We are then faced with the question of selecting which p shifts to apply and whether we should consider applying more than p shifts.

For example, m shifts can be applied until a Ritz pair satisfies the convergence tolerance. The Ritz pairs can then be deflated (or locked). (This is equivalent to the *deflated iterative Arnoldi* algorithm given by Saad [24, p. 181] and used in the implementations in [2, 27].) This approach allows us to implicitly apply a polynomial filter of the maximum degree. (Application of more than $r + p$ shifts will require applying explicit polynomials in $\mathbf{A}$.) However, as more shifts are applied, the cost in computing the subsequent Arnoldi reduction increases.

A strategy that varies r, p (relative to k) and the shifts used during every iteration will give the best results. This is the subject of current research. The recent report [30] discusses an adaptive strategy for symmetric eigenvalue problems. A near-optimal adaptive strategy should be possible because of the connection between bIRAM and subspace iteration [14].

5.3. Convergence. Suppose that $(\mathbf{s}, \theta)$ is an eigenpair of $\mathbf{H}_m$. It follows easily from Equation (3.1) that

$$\mathbf{A}\mathbf{V}_m\mathbf{s} - \mathbf{V}_m\mathbf{s}\theta = \mathbf{F}_m\mathbf{E}_m^T\mathbf{s}, \tag{5.2}$$

and so $\|\mathbf{AV}_m\mathbf{s} - \mathbf{V}_m\mathbf{s}\theta\| = \|\mathbf{F}_m\|\,\|\mathbf{E}_m^T\mathbf{s}\| = \|\mathbf{G}_{m+1,m}\|\,\|\mathbf{E}_m^T\mathbf{s}\|$. Thus, if the last b components of $\mathbf{s}$ are small relative to the size of $\|\mathbf{G}_{m+1,m}\|$, then the Ritz pair $(\mathbf{z} = \mathbf{V}_m\mathbf{s}, \theta)$ is an exact eigenpair for a matrix near $\mathbf{A}$. This follows since (5.2) may be rewritten as $(\mathbf{A} - \mathbf{F}_m\mathbf{E}_m^T\mathbf{s}\mathbf{z}^H)\mathbf{z} = \mathbf{z}\theta$.

The iteration in bIRAM terminates at the value of i when the k wanted eigenvalues of $\mathbf{H}_m^{(i)}$ satisfy (5.2). The eigenvalues are partitioned as in (5.1) so that the wanted ones correspond to the eigenvalues of $\mathbf{A}$ desired.

5.4. Deflation. Because a bIRAM may be viewed as a truncation of the standard implicitly shifted QR-iteration, it shares a number of the QR-iteration's desirable properties. The main advantages of a numerically stable deflation strategy are the reduction of the working size of the reduction and the ability to determine clusters of nearby eigenvalues without requiring the block size to greater than or equal to the size of the cluster. The deflation scheme developed for the bIRAM implementation is an extension of the techniques discussed by Lehoucq and Sorensen [15].

Let $\mathbf{H}$ be an unreduced band upper Hessenberg matrix of order m with block size b. Then, given a Ritz value θ, we construct an orthonormal matrix $\mathbf{Q}$ such that $\mathbf{Qs} = \mathbf{e}_1$ and

$$\mathbf{E}_m^T\mathbf{Q} = \left[\begin{array}{cccc} \underbrace{\mathbf{E}_m^T\mathbf{s}\;0}_{b\text{ columns}} & \mathbf{0} & \cdots & \mathbf{C} \end{array}\right],$$

where $\mathbf{C}$ is an upper triangular matrix of order b. This specification for $\mathbf{Q}$ is more restrictive than that originally proposed in [15], where $\mathbf{Q}$ needed to satisfy only $\mathbf{Qs} = \mathbf{e}_1$ and $\|\mathbf{E}_m^T\mathbf{Q}\| = \mathcal{O}(\|\mathbf{E}_m^T\mathbf{s}\|)$. Numerical experience indicates that this restrictive $\mathbf{Q}$ is more robust in practice.

Applying this orthonormal matrix $\mathbf{Q}$ to the Arnoldi reduction gives

$$\mathbf{AVQ} = \mathbf{VQ}\begin{bmatrix} \theta & \bar{\mathbf{h}}^T \\ 0 & \bar{\mathbf{H}} \end{bmatrix} + \mathbf{FE}_m^T\mathbf{Q}. \tag{5.3}$$

The first column of the previous equation gives the relationship

$$\mathbf{Az}_1 = \mathbf{z}_1\theta + \mathbf{FE}_m^T\mathbf{s}.$$

The Ritz vector $\mathbf{z}_1 = \mathbf{V}_m\mathbf{s}$ is considered *locked*, and subsequent implicit restarting is performed on $\widetilde{\mathbf{V}}_m$ consisting of the last $m\cdot b-1$ columns of $\mathbf{VQ}$. Equation (5.3) is not a block Arnoldi reduction because $\bar{\mathbf{H}}$ no longer is a band Hessenberg matrix. However, a sequence of elementary unitary matrices may be applied without disturbing the structure of $\mathbf{FCE}_m^T$ and the first column of $\mathbf{Q}^T\mathbf{HQ}$. See [15] for details.

This deflation process allows us to incrementally build an approximate partial Schur decomposition $\mathbf{AZ}_j \approx \mathbf{Z}_j\mathbf{T}_j$ that satisfies

$$\|\mathbf{AZ}_j - \mathbf{Z}_j\mathbf{T}_j\| = \mathcal{O}(\|\mathbf{F}_m\|\,\|\mathbf{E}_m^T\mathbf{s}\|).$$

Here $\mathbf{T}_j$ is an upper triangular matrix of order j containing the Ritz values of interest. When $\mathbf{A}$ is Hermitian, $\mathbf{T}_j$ is a diagonal matrix. See [15, pp. 801–802] for locking complex conjugate pairs of Ritz values in real arithmetic so that complex arithmetic can be avoided for real $\mathbf{A}$.

A significant benefit of this deflation procedure is that as Ritz values converge they can be implicitly deflated from the active reduction without the need to build a new reduction from scratch. This avoids additional matrix-vector products and orthogonalization costs that would be incurred if the deflation was performed explicitly.

In terms of the current reduction we have

$$\mathbf{A}\begin{bmatrix} \mathbf{Z}_j & \mathbf{V}_m \end{bmatrix} = \begin{bmatrix} \mathbf{Z}_j & \mathbf{V}_m \end{bmatrix} \begin{bmatrix} \mathbf{T}_j & \mathbf{M}_j \\ \mathbf{0} & \mathbf{H}_m \end{bmatrix} + \mathbf{F}_m \mathbf{E}_{m+j}^T.$$

Here $\mathbf{Z}_j$ contains the j *locked* approximate Schur vectors. We define the *active* reduction to be

$$\hat{\mathbf{A}}\mathbf{V}_m = \mathbf{V}_m \mathbf{H}_m + \mathbf{F}_m \mathbf{E}_m^T, \tag{5.4}$$

where $\hat{\mathbf{A}} = (\mathbf{I} - \mathbf{Z}_m \mathbf{Z}_m^H)\mathbf{A}$. Note that the projector $\mathbf{I} - \mathbf{Z}_m \mathbf{Z}_m^H$ is never applied. Instead, as $\mathbf{V}_m$ is computed, orthogonality of its columns against $\mathbf{Z}_j$ is maintained through the coefficients $\mathbf{M}_j$. This achieves a clean form of selective orthogonalization [19].

An implementation detail is the selection of an appropriate locking criterion. For the numerical experiments, we used a locking tolerance of

$$\|\mathbf{E}_m \mathbf{s}_i\| \le \max(10^{-2}\,\eta_U,\,\epsilon_M)$$

where η_U is the user-specified tolerance and ϵ_M is the machine precision.

An algorithmic issue that arises is how to handle situations where the number of converged vectors to be locked is not a multiple of the block size. At the completion of the deflation procedure, the active reduction has an incomplete last block. To simplify the implicit restarting mechanism, we fill out this last block so that the active reduction is of length $m \cdot b$.

6. Numerical Experiments. In this section we compare results for the blocked and unblocked methods. Our aim is to acquire a better understanding of the practical behavior of these methods. Comparisons also are made to other block Arnoldi methods to demonstrate the effectiveness of implicit restarting.

Experiments were conducted in MATLAB on a Sun SPARCstation20 using IEEE double-precision arithmetic. When comparing the different algorithms and various block sizes, we fix the maximum number of vectors $b \cdot m$ in the subspace where $m = r + p$. During every iteration the number of shifts applied is p. With these requirements the orthogonalization costs per iteration remain approximately constant. Since this is only an experimental MATLAB code, we rely on the iteration count and the total number of matrix-vector products to rank the performance of the methods. For the matrix-vector products we provide both an individual count and the number of times the matrix is applied to a block of vectors. In many instances, the cost of computing a few matrix vector products is commensurate with that of one matrix-vector product, and this primarily determines the efficiency. Inherent in the number of iterations is the cost associated with the solution of a small eigenvalue problem and the QR iteration associated with the implicit restart mechanism.

An eigenpair $(\theta, \mathbf{s})$ of $\mathbf{H}_m$ is accepted as converged if

$$\|\mathbf{E}_m^T \mathbf{s}\|\,\|\mathbf{G}_{m+1,m}\| \le |\theta|\,\eta_U,$$

where η_U is a user-specified tolerance. Each pair that satisfies the convergence tolerance is then deflated by using the techniques of § 5.4, resulting in a partial Schur decomposition.

TABLE 6.1

2-D Laplacian ($\rho = 0$) matrix of order 2500. Results are for finding the six eigenvalues of smallest magnitude using a tolerance of 10^{-10} and a maximum subspace dimension of $m \cdot b = 24$. The implicit restarting strategy used was to apply the p least wanted Ritz values as shifts.

b	r	p	m	Iters.	MV Products	$\|\|\mathbf{AZ}_6 - \mathbf{Z}_6\mathbf{T}_6\|\|_2$
1	6	18	24	41	608 (608)	$\mathcal{O}(10^{-12})$
1	6	18	24	41	608 (608)	$\mathcal{O}(10^{-12})$
2	2	10	12	35	694 (347)	$\mathcal{O}(10^{-12})$
2	3	9	12	30	540 (270)	$\mathcal{O}(10^{-12})$
3	2	6	8	42	744 (248)	$\mathcal{O}(10^{-12})$
4	2	4	6	49	800 (200)	$\mathcal{O}(10^{-12})$
6	1	3	4	58	1068 (178)	$\mathcal{O}(10^{-12})$
8	1	2	3	82	1344 (168)	$\mathcal{O}(10^{-12})$

TABLE 6.2

2-D Laplacian ($\rho = 0$) matrix of order 2500. Results are for finding the six eigenvalues of smallest magnitude using a tolerance of 10^{-6} and a maximum subspace dimension of $m \cdot b = 24$. The implicit restarting strategy used was to apply the p least wanted Ritz values as shifts.

b	r	p	m	Iters.	MV Products	$\|\|\mathbf{AZ}_6 - \mathbf{Z}_6\mathbf{T}_6\|\|_2$
1	6	18	24	35	509 (509)	$\mathcal{O}(10^{-8})$
2	2	10	12	21	420 (210)	$\mathcal{O}(10^{-8})$
2	3	9	12	24	434 (217)	$\mathcal{O}(10^{-9})$
3	2	6	8	30	543 (181)	$\mathcal{O}(10^{-8})$
4	2	4	6	40	656 (164)	$\mathcal{O}(10^{-9})$
6	1	3	4	44	804 (134)	$\mathcal{O}(10^{-8})$

6.1. Model Problem. The first problem for which we present results is a two-dimensional convection–diffusion problem

$$-\Delta u + \rho(u_x + u_y) = \lambda u.$$

Here we discretize using centered finite differences on the unit square with zero boundary data. We have chosen this example because it has the following interesting properties:

- Eigenvalues and eigenvectors of the resulting matrix are known explicitly.
- Many eigenvalues have multiplicity two.
- As the mesh size decreases, relative separation of all the eigenvalues decreases. All are contained within the interval $(0,\ 8)$.
- As ρ increases, so does the non-normality of the matrix.

Although block algorithms are not always required for determining multiplicities, they require fewer applications of $\mathbf{A}$ than does the unblocked variant. Block methods demonstrate group convergence to multiple eigenvalues, whereas an unblocked method requires a Ritz value and vector to be fairly well approximated before any additional copies emerge.

6.2. Two-Dimensional Laplacian. Tables 6.1 and 6.2 list the results for finding the six eigenvalues of smallest magnitude and corresponding eigenvectors for the two-dimensional Laplacian ($\rho = 0$) matrix of order 2500 for convergence tolerances of 10^{-10} and 10^{-6}, respectively. A moderate maximum subspace dimension of $m \cdot b = 24$ was used with an exact shifting strategy where the p least wanted Ritz values (relative to magnitude-based ordering) were used as shifts. Although when $b > 1$ the number

TABLE 6.3

2-D Laplacian ($\rho = 0$) matrix of order 2500. Results are for finding the six eigenvalues of smallest magnitude using a tolerance of 10^{-10} and a maximum subspace dimension of $m \cdot b = 36$. The implicit restarting strategy used was to apply the p least wanted Ritz values as shifts.

b	r	p	m	Iters.	MV Products		$\|\|\mathbf{A}\mathbf{Z}_6 - \mathbf{Z}_6\mathbf{T}_6\|\|_2$
1	6	30	36	21	540	(540)	$\mathcal{O}(10^{-13})$
2	3	15	18	65	1916	(958)	$\mathcal{O}(10^{-13})$
3	2	10	12	109	3171	(1057)	$\mathcal{O}(10^{-12})$
4	2	7	9	71	2004	(501)	$\mathcal{O}(10^{-12})$
6	1	5	6	67	2040	(340)	$\mathcal{O}(10^{-12})$
9	1	3	4	49	1341	(149)	$\mathcal{O}(10^{-12})$

TABLE 6.4

2-D Laplacian ($\rho = 0$) matrix of order 2500. Results are for finding the six eigenvalues of smallest magnitude using a tolerance of 10^{-6} and a maximum subspace dimension of $m \cdot b = 36$. The implicit restarting strategy used was to apply the p least wanted Ritz values as shifts.

b	r	p	m	Iters.	MV Products		$\|\|\mathbf{A}\mathbf{Z}_6 - \mathbf{Z}_6\mathbf{T}_6\|\|_2$
1	6	30	36	18	483	(483)	$\mathcal{O}(10^{-9})$
2	3	15	18	17	514	(257)	$\mathcal{O}(10^{-8})$
3	2	10	12	28	846	(282)	$\mathcal{O}(10^{-8})$
4	2	7	9	21	604	(151)	$\mathcal{O}(10^{-8})$
6	1	5	6	28	852	(142)	$\mathcal{O}(10^{-9})$
9	1	3	4	49	1026	(114)	$\mathcal{O}(10^{-8})$

of Ritz values we are able to filter out is less than the total number of unwanted Ritz values (in this case 18), we still obtain good convergence to the eigenvalues of interest. Here the benefit of group convergence outweighs any possible degradation in performance due to a smaller value of p.

Tables 6.3 and 6.4 show the results for the same matrix using a larger subspace $m \cdot b = 36$ with the same exact shifting strategy described above. By using a larger subspace we gain the advantages of a longer recurrence in the Arnoldi process. However, a larger subspace also translates into an increase in the number of unwanted eigenvalues that cannot be used as shifts. For instance, in the previous example where we used a maximum basis of 24 and $b = 2$, we are able to apply 9 of the 18 unwanted eigenvalues. Using a maximum basis of 36 and $b = 2$, we can apply only 15 of 30 unwanted eigenvalues. We observed in our experiments that when a large subspace is used relative to the number of eigenvalues desired, the exact shifting strategy was not necessarily the best choice. This situation was especially evident for smaller values of η_U.

If we instead use the roots of a Chebyshev polynomial of degree p defined on the interval that excludes the six smallest eigenvalues of $\mathbf{H}_m$ as shifts, a better filter results for small block sizes. For larger values of b, the degree of the Chebyshev polynomial p is not sufficient to adequately characterize the particular region of the spectrum which we would like to filter. When the maximum number $b \cdot m$ of vectors is small (say less than 10), the use of Leja shifts gives the best results [2].

6.2.1. Two-Dimensional Convection-Diffusion Matrix. Tables 6.7 and 6.8 list the results for finding the six eigenvalues of smallest magnitude and the corresponding Schur vectors for the two-dimensional convection-diffusion matrix with $\rho = 20$. Again by using a moderate maximum subspace dimension we get promising results for $b > 1$. For $\eta_U = 10^{-6}$, the number of applications of $\mathbf{A}$ decreases as the

TABLE 6.5

2-D Laplacian ($\rho = 0$) matrix of order 2500. Results are for finding the six eigenvalues of smallest magnitude using a tolerance of 10^{-10} and a maximum subspace dimension of $m \cdot b = 36$. The implicit restarting strategy used was to apply the roots of a Chebyshev polynomial of degree p as shifts.

b	r	p	m	Iters.	MV Products		$\|\|\mathbf{AZ}_6 - \mathbf{Z}_6\mathbf{T}_6\|\|_2$
1	6	30	36	21	569	(569)	$\mathcal{O}(10^{-13})$
2	3	15	18	29	866	(433)	$\mathcal{O}(10^{-13})$
3	2	10	12	47	1410	(470)	$\mathcal{O}(10^{-12})$
4	2	7	9	62	1740	(435)	$\mathcal{O}(10^{-12})$
6	1	5	6	142	4290	(715)	$\mathcal{O}(10^{-12})$

TABLE 6.6

2-D Laplacian ($\rho = 0$) matrix of order 2500. Results are for finding the six eigenvalues of smallest magnitude using a tolerance of 10^{-6} and a maximum subspace dimension of $m \cdot b = 36$. The implicit restarting strategy used was to apply the roots of a Chebyshev polynomial of degree p as shifts.

b	r	p	m	Iters.	MV Products		$\|\|\mathbf{AZ}_6 - \mathbf{Z}_6\mathbf{T}_6\|\|_2$
1	6	30	36	18	483	(483)	$\mathcal{O}(10^{-9})$
2	3	15	18	20	598	(299)	$\mathcal{O}(10^{-8})$
3	2	10	12	31	930	(310)	$\mathcal{O}(10^{-8})$
4	2	7	9	43	1216	(304)	$\mathcal{O}(10^{-8})$
6	1	5	6	93	2820	(470)	$\mathcal{O}(10^{-8})$

block size increases. This trend is also displayed for $\eta_U = 10^{-10}$. But this smaller tolerance also requires a significant number of additional applications of $\mathbf{A}$ as the blocksize increases.

6.2.2. Matrices from Harwell-Boeing Collection. Other matrices that we have used to evaluate the potential of implicit restarting for block methods are taken from the Harwell-Boeing collection [8] of sparse matrices. These problems were chosen because they have appeared in the literature for sparse nonsymmetric eigenvalue problems. For these matrices we used $\eta_U = 10^{-12}$.

Tables 6.9 and 6.10 show the results for the matrix GRE1107. This matrix, taken from a simulation study for a computer system, has a dimension of 1107 with 5664 nonzero entries. The matrix is real nonsymmetric. Here we are able to directly compare our results with another block Arnoldi implementation [27] EB13. We include this comparison to emphasize the potential of implicit restarting for block Arnoldi methods. In all cases we compare our results with the "best" results obtained from EB13. EB13 provides both blocked and unblocked Arnoldi methods and includes options for Chebyshev acceleration and Chebyshev preconditioning. Restarting for EB13 is done explicitly using linear combinations of approximate Schur vectors. Comparisons are made for maximum subspace dimensions of 24 and 40.

For GRE1107 the best results from EB13 were obtained using the unblocked Arnoldi method with Chebyshev preconditioning. The preconditioned method exhibits very fast convergence (2 iterations); but because the operator used to generate the Arnoldi reduction is a high degree polynomial in $\mathbf{A}$, this approach requires a large number of matrix-vector products. (Implicit restarting is performed by using the p least wanted Ritz values as shifts.) bIRAM requires substantially fewer matrix-vector products regardless of the blocksize.

Sadkane [26] also presents numerical results using a block Arnoldi method for

TABLE 6.7

2-D convection-diffusion matrix of order 2500, $\rho = 20$. Results are for finding the six eigenvalues of smallest magnitude using a tolerance of 10^{-10} and a maximum subspace dimension of $m \cdot b = 24$. The implicit restarting strategy used was to apply the p least wanted Ritz values as shifts.

b	r	p	m	Iters.	MV Products		$\|\|\mathbf{AZ}_6 - \mathbf{Z}_6\mathbf{T}_6\|\|_2$
1	6	18	24	34	530	(530)	$\mathcal{O}(10^{-12})$
2	3	9	12	34	610	(305)	$\mathcal{O}(10^{-12})$
2	4	8	12	34	544	(272)	$\mathcal{O}(10^{-11})$
3	2	6	8	40	720	(240)	$\mathcal{O}(10^{-12})$
4	2	4	6	47	760	(190)	$\mathcal{O}(10^{-10})$
6	1	3	4	65	1182	(197)	$\mathcal{O}(10^{-11})$

TABLE 6.8

2-D convection-diffusion matrix of order 2500, ρ=20. Results are for finding the six eigenvalues of smallest magnitude using a tolerance of 10^{-6} and a maximum subspace dimension of $m \cdot b = 24$. The implicit restarting strategy used was to apply the p least wanted Ritz values as shifts.

b	r	p	m	Iters.	MV Products		$\|\|\mathbf{AZ}_6 - \mathbf{Z}_6\mathbf{T}_6\|\|_2$
1	6	18	24	30	445	(445)	$\mathcal{O}(10^{-8})$
2	2	10	12	22	444	(222)	$\mathcal{O}(10^{-8})$
2	3	9	12	26	462	(231)	$\mathcal{O}(10^{-8})$
3	2	6	8	30	528	(176)	$\mathcal{O}(10^{-7})$
4	2	4	6	38	616	(154)	$\mathcal{O}(10^{-7})$

TABLE 6.9

GRE_1107, Harwell-Boeing collection. Results are for finding the eight rightmost eigenvalues using a tolerance of 10^{-12} and a maximum subspace dimension of $m \cdot b = 24$. The implicit restarting strategy used for bIRAM was to use the p least wanted Ritz values as shifts. A3 uses the Chebyshev preconditioned Arnoldi option of EB13.

	b	r	p	m	Iters.	MV Prod.		$\|\|\mathbf{AZ}_8 - \mathbf{Z}_8\mathbf{T}_8\|\|_2$
IRA	1	8	16	24	33	448	(448)	$\mathcal{O}(10^{-15})$
	2	4	8	12	37	580	(290)	$\mathcal{O}(10^{-13})$
	3	2	6	8	40	720	(240)	$\mathcal{O}(10^{-12})$
	4	2	4	6	62	980	(245)	$\mathcal{O}(10^{-12})$
	8	1	2	3	101	1664	(208)	$\mathcal{O}(10^{-13})$
A3	1			24	2	984	(984)	$\mathcal{O}(10^{-11})$

TABLE 6.10

GRE_1107, Harwell-Boeing collection. Results are for finding the eight rightmost eigenvalues using a tolerance of 10^{-12} and a maximum subspace dimension of $m \cdot b = 40$. The implicit restarting strategy used for bIRAM was to use the p least wanted Ritz values as shifts. A3 and AB3 use unblocked and blocked Chebyshev preconditioned Arnoldi options, respectively, available in EB13.

	b	r	p	m	Iters.	MV Prod.		$\|\|\mathbf{AZ}_8 - \mathbf{Z}_8\mathbf{T}_8\|\|_2$
IRA	1	8	32	40	15	392	(392)	$\mathcal{O}(10^{-13})$
	2	4	16	20	16	506	(253)	$\mathcal{O}(10^{-13})$
	4	2	8	10	24	788	(197)	$\mathcal{O}(10^{-12})$
	10	1	3	4	55	1710	(171)	$\mathcal{O}(10^{-12})$
A3	1			40	2	1656	(1656)	$\mathcal{O}(10^{-12})$
AB3	10			4	3	4940	(494)	$\mathcal{O}(10^{-12})$
AC	8			320		2920	(365)	$\mathcal{O}(10^{-9})$
DAV	8			320		3464	(433)	$\mathcal{O}(10^{-9})$

TABLE 6.11

HOR131, (N=434) Harwell-Boeing Collection. Results are for finding the eight rightmost eigenvalues using a tolerance of 10^{-12} and a maximum subspace dimension of $m \cdot b = 24$. The implicit restarting strategy used for bIRAM was to use the p least wanted Ritz values as shifts. A1 and A2 use a unblocked Arnoldi and Arnoldi with Chebyshev acceleration options, respectively, available in EB13.

	b	r	p	m	Iters.	MV Prod.	$\|\|\mathbf{AZ}_8 - \mathbf{Z}_8\mathbf{T}_8\|\|_2$
IRA	1	8	16	24	5	77 (77)	$\mathcal{O}(10^{-15})$
	2	4	8	12	5	84 (42)	$\mathcal{O}(10^{-14})$
	3	3	5	8	6	99 (33)	$\mathcal{O}(10^{-15})$
	4	2	4	6	6	108 (27)	$\mathcal{O}(10^{-14})$
	8	1	2	3	10	176 (22)	$\mathcal{O}(10^{-15})$
A1	1			24	22	497 (497)	$\mathcal{O}(10^{-13})$
A2	1			24	11	439 (439)	$\mathcal{O}(10^{-13})$

TABLE 6.12

HOR131, (N=434) Harwell-Boeing Collection. Results are for finding the eight rightmost eigenvalues using a tolerance of 10^{-12} and a maximum subspace dimension of $m \cdot b = 40$. The implicit restarting strategy used for bIRAM was to use the p least wanted Ritz values as shifts. A1 and AB2 use a Arnoldi and block Arnoldi with Chebyshev acceleration options available in EB13.

	b	r	p	m	Iters.	MV Prod.	$\|\|\mathbf{AZ}_8 - \mathbf{Z}_8\mathbf{T}_8\|\|_2$
IRA	1	8	32	40	2	71 (71)	$\mathcal{O}(10^{-15})$
	2	4	16	20	2	72 (36)	$\mathcal{O}(10^{-15})$
	4	2	8	10	3	104 (26)	$\mathcal{O}(10^{-15})$
	10	1	3	4	6	190 (19)	$\mathcal{O}(10^{-15})$
A1	1			40	5	202 (202)	$\mathcal{O}(10^{-12})$
AB2	10			4	3	414 (42)	$\mathcal{O}(10^{-12})$
DAV	8			320		264 (33)	$\mathcal{O}(10^{-7})$

GRE1107. Here he uses explicit restarting (the polynomial filter is applied with the matrix) with Chebyshev acceleration using polynomials of degree 40 and a maximum subspace dimension of 40·8. We have appended his results for block Arnoldi-Chebyshev (AC) and Davidson (DAV) methods to the bottom of Table 6.10 for reference. It should be noted that the results for bIRAM and EB13 are for a maximum subspace of 40 with a tolerance of $\eta_U = 10^{-12}$.

Tables 6.11 and 6.12 show the results for the matrix HOR131. This matrix, taken from a flow network problem, is of order 434 with 4182 nonzero entries. The matrix is real nonsymmetric with symmetric structure. The performance of bIRAM is superior when compared with EB13 and the unsymmetric Davidson method. (The results for the unsymmetric Davidson implementation used the diagonal of the matrix as the preconditioner.) One interesting observation is that by using implicit restarting we no longer require large subspace dimensions to take advantage of block methods. Other Arnoldi methods have typically required longer Arnoldi reductions in order to achieve reasonable convergence. The ability to use smaller basis sets reduces the relative cost of maintaining full orthogonality of the Arnoldi vectors.

It is clear from our numerical results for GRE1107 and HOR131 that implicit restarting represents a substantial improvement over other implementations of block Arnoldi methods.

Figure 6.13 show the results for the matrix NNC1374. This matrix, derived from a model of an advanced gas-cooled reactor core, is of order 1374 with 8588 nonzero

TABLE 6.13

NNC1374, Harwell-Boeing Collection. Results are for finding the eight eigenvalues of largest magnitude using a tolerance of 10^{-12} and a maximum subspace dimension of $m \cdot b = 24$. The implicit restarting strategy used for bIRAM was to use the p least wanted Ritz values as shifts.

	b	r	p	m	Iters.	MV Prod.	$\|\|\mathbf{AZ}_8 - \mathbf{Z}_8\mathbf{T}_8\|\|_2$
IRA	1	8	16	24	16	235 (235)	$\mathcal{O}(10^{-13}$
	2	4	8	12	18	286 (143)	$\mathcal{O}(10^{-12}$
	3	2	6	9	20	357 (119)	$\mathcal{O}(10^{-14}$
	4	1	5	6	23	464 (116)	$\mathcal{O}(10^{-13}$
	4	2	4	6	28	456 (114)	$\mathcal{O}(10^{-13}$
	8	1	2	3	62	1016 (127)	$\mathcal{O}(10^{-13}$

entries. The matrix is real nonsymmetric.

7. Conclusions. This article investigated an implicitly restarted block Arnoldi method. The two benefits of increasing the block size used with an implicitly block Arnoldi method are

- the ability to reliably compute multiple and/or clustered eigenvalues while using a modest value of η_U, and
- the number of applications of $\mathbf{A}$ tends to decreases as the block size increases. This will result in a substantial increase in efficiency if applying $\mathbf{A}$ to a group of vectors is faster than applying $\mathbf{A}$ to a vector one at a time.

Our experimental results also indicate that an implicitly restarted scheme is superior to other block methods that have appeared in the literature. Not only does bIRAM use dramatically fewer matrix-vector products, it also requires significantly fewer Arnoldi vectors to be stored.

REFERENCES

[1] J. Baglama, D. Calvetti, and L. Reichel, *Iterative methods for the computation of a few eigenvalues of a large symmetric matrix*, BIT, 36 (1996), pp. 400–440.

[2] J. Baglama, D. Calvetti, L. Reichel, and A. Ruttan, *Computation of a few close eigenvalues of a large matrix with application to liquid crystal modeling*, technical report, Department of Mathematics and Computer Science, Kent State University, 1997.

[3] F. Chatelin, *Eigenvalues of Matrices*, Wiley, 1993.

[4] J. Cullum and W. E. Donath, *A block Lanczos algorithm for computing the q algebraically largest eigenvalues and a corresponding eigenspace for large, sparse symmetric matrices*, in Proceedings of the 1974 IEEE Conference on Decision and Control, New York, 1974, pp. 505–509.

[5] 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 (1976), pp. 772–795.

[6] J. J. Dongarra, J. DuCroz, I. S. Duff, and S. Hammarling, *A set of Level 3 Basic Linear Algebra Subprograms*, ACM Transactions on Mathematical Software, 16 (1990), pp. 1–17.

[7] J. J. Dongarra, J. DuCroz, S. Hammarling, and R. J. Hanson, *An extended set of Fortran Basic Linear Algebra Subprograms*, ACM Trans. on Math. Software, 14 (1988), pp. 1–17.

[8] I. S. Duff, R. G. Grimes, and J. G. Lewis, *Users' guide for the Harwell-Boeing sparse matrix collection (Release 1)*, Tech. Rep. RAL-92-086, Rutherford Appleton Laboratory, Chilton, England, 1992.

[9] G. H. Golub and C. F. V. Loan, *Matrix Computations*, Johns Hopkins University Press, Baltimore, second ed., 1989.

[10] ———, *Matrix Computations*, Johns Hopkins University Press, Baltimore, third ed., 1996.

[11] G. H. Golub and R. Underwood, *The block Lanczos method for computing eigenvalues*, in Mathematical Software III, J. R. Rice, ed., Academic Press, New York, 1977, 1977, pp. 361–377.

[12] R. G. Grimes, J. G. Lewis, and H. D. Simon, *A shifted block Lanczos algorithm for solving sparse symmetric generalized eigenproblems*, SIAM J. Matrix Analysis and Applications, 15 (1994), pp. 228–272.

[13] W. Karush, *An iterative method for finding characteristics vectors of a symmetric matrix*, Pacific J. Mathematics, 1 (1951), pp. 233–248.

[14] R. B. Lehoucq, *Truncated QR algorithms and the numerical solution of large scale eigenvalue problems*, Preprint MCS-P648-0297, Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, Ill, 1997.

[15] R. B. Lehoucq and D. C. Sorensen, *Deflation techniques for an implicitly restarted Arnoldi iteration*, SIAM J. Matrix Analysis and Applications, 17 (1996), pp. 789–821.

[16] R. B. Lehoucq, D. C. Sorensen, and C. Yang, *ARPACK USERS GUIDE: Solution of large scale eigenvalue problems by implicitly restarted Arnoldi methods*, Tech. Rep. CRPC-TR96665, Center for Research on Parallel Computation, Rice University, Houston, 1996.

[17] R. B. Morgan, *Computing interior eigenvalues of large matrices*, Linear Algebra and Its Applications, 154/156 (1991), pp. 289–309.

[18] C. C. Paige, B. N. Parlett, and H. A. van der Vorst, *Approximate solutions and eigenvalue bounds from Krylov subspaces*, Numerical Linear Algebra with Applications, 2 (1995), pp. 115–134.

[19] B. N. Parlett and D. Scott, *The Lanczos algorithm with selective orthogonalization*, Mathematics of Computation, 33 (1979), pp. 217–238.

[20] A. Ruhe, *Implementation aspects of band Lanczos algorithms for computation of eigenvalues of large sparse symmetric matrices*, Mathematics of Computation, 33 (1979), pp. 680–687.

[21] Y. Saad, *On the rates of convergence of the Lanczos and the block Lanczos methods*, SIAM Journal of Numerical Analysis, 17 (1980), pp. 687–706.

[22] ———, *Variations on Arnoldi's method for computing eigenelements of large unsymmetric matrices*, Linear Algebra and Its Applications, 34 (1980), pp. 269–295.

[23] ———, *Projection methods for solving large sparse eigenvalue problems*, in Matrix Pencil Proceedings, B. Kågström and A. Ruhe, eds., vol. 973 of Lecture Notes in Mathematics, Springer-Verlag, Berlin, 1982, pp. 121–144.

[24] ———, *Numerical Methods for Large Eigenvalue Problems*, Halsted Press, 1992.

[25] M. Sadkane, *A block Arnoldi–Chebyshev method for computing the leading eigenpairs of large sparse unsymmetric matrices*, Numerische Mathematik, 64 (1993), pp. 181–193.

[26] ———, *Block Arnoldi-Chebychev and Davidson methods for unsymmetric large eigenvalue problems*, Numerische Mathematik, 64 (1993), pp. 181–193.

[27] J. A. Scott, *An Arnoldi code for computing selected eigenvalues of sparse real unsymmetric matrices*, ACM Trans. Mathematical Software, 21 (1995), pp. 432–475.

[28] G. L. G. Sleijpen and H. van der Vorst, *A Jacobi-Davidson iteration method for linear eigenvalue problems*, SIAM J. Matrix Analysis and Applications, 17 (1996), pp. 401–425.

[29] D. C. Sorensen, *Implicit application of polynomial filters in a k-step Arnoldi method*, SIAM J. Matrix Analysis and Applications, 13 (1992), pp. 357–385.

[30] A. Stathopoulos, Y. Saad, and K. Wu, *Dynamic thick restarting of the Davidson, and the implicitly restarted Arnoldi methods*, SIAM J. Scientific Computing, (1997). To appear.