

ADAPTIVELY PRECONDITIONED GMRES ALGORITHMS

J. BAGLAMA ^{*}, D. CALVETTI [†], G. H. GOLUB [‡], AND L. REICHEL [§]

Abstract. The restarted GMRES algorithm proposed by Saad and Schultz [22] is one of the most popular iterative methods for the solution of large linear systems of equations $Ax = b$ with a nonsymmetric and sparse matrix. This algorithm is particularly attractive when a good preconditioner is available. The present paper describes two new methods for determining preconditioners from spectral information gathered by the Arnoldi process during iterations by the restarted GMRES algorithm. These methods seek to determine an invariant subspace of the matrix A associated with eigenvalues close to the origin, and move these eigenvalues so that a higher rate of convergence of the iterative methods is achieved.

Key words. iterative method, nonsymmetric linear system, Arnoldi process

AMS subject classifications. 65F10

1. Introduction. Many problems in Applied Mathematics and Engineering give rise to very large linear systems of equations

$$(1.1) \quad Ax = b, \quad A \in \mathbf{R}^{n \times n}, \quad x, b \in \mathbf{R}^n,$$

with a sparse nonsymmetric nonsingular matrix A . It is often desirable, and sometimes necessary, to solve these systems by an iterative method. Let x_0 be an initial approximate solution of (1.1), and let $r_0 = b - Ax_0$ be the associated residual vector. Introduce the Krylov subspaces

$$(1.2) \quad \mathbf{K}_m(A, r_0) = \text{span}\{r_0, Ar_0, \dots, A^{m-1}r_0\}, \quad m = 1, 2, \dots,$$

associated with the matrix A and vector r_0 . Many popular iterative methods determine the m th iterate, x_m , so that $x_m - x_0 \in \mathbf{K}_m(A, r_0)$. We refer to such methods as Krylov subspace iterative methods; see, e.g., Freund et al. [12] for a recent review.

Let the iterate x_m be generated by a Krylov subspace iterative method. Then the residual error $r_m = b - Ax_m$ associated with x_m satisfies

$$(1.3) \quad r_m = p_m(A)r_0,$$

where the residual polynomial p_m is determined by the iterative method, and satisfies $p_m(0) = 1$. Let $\|\cdot\|$ denote the Euclidean norm on \mathbf{R}^n , as well as the associated induced matrix norm on $\mathbf{R}^{n \times n}$. The restarted Generalized Minimal Residual (GMRES(m)) algorithm by Saad and Schultz [22], described in Section 3, is one of the most popular Krylov subspace iterative methods for the solution of linear systems with a nonsymmetric matrix. The residual polynomial determined by this algorithm satisfies

$$(1.4) \quad \|p_m(A)r_0\| = \min_{p \in \Pi_m^0} \|p(A)r_0\|,$$

^{*} Department of Mathematics and Computer Science, Kent State University, Kent, OH 44242. E-mail: jbaglama@mcs.kent.edu. Research supported in part by NSF grant F377 DMR-8920147 ALCOM.

[†] Department of Mathematical Sciences, Stevens Institute of Technology, Hoboken, NJ 07030. E-mail: na.calvetti@na-net.ornl.gov. Research supported in part by NSF grant DMS-9404692.

[‡] Computer Science Department, Stanford University, Stanford, CA 94305.

[§] Department of Mathematics and Computer Science, Kent State University, Kent, OH 44242. E-mail: reichel@mcs.kent.edu. Research supported in part by NSF grant DMS-9404706.

where Π_m^0 denotes the set of all polynomials p of degree at most m such that $p(0) = 1$.

The analysis and implementation of the restarted GMRES(m) algorithm, and modifications thereof, continue to receive considerable attention; see, e.g., [4, 5, 7, 8, 9, 10, 14, 25, 26]. These algorithms are particularly attractive when a suitable preconditioner $M^{-1} \in \mathbf{R}^{n \times n}$ for the matrix A is available; see, e.g., [2, 15, 21] for recent discussions on preconditioners. A matrix M^{-1} is a good preconditioner if the application of an iterative method of interest to the preconditioned linear system of equations

$$(1.5) \quad M^{-1}Ax = M^{-1}b$$

gives a higher rate of convergence of the computed iterates than application of the iterative method to the original linear system (1.1). Moreover, we would like the preconditioner M^{-1} have the property that for any $w \in \mathbf{R}^n$, the vector $M^{-1}w$ can be rapidly evaluated. The matrix M^{-1} in (1.5) is sometimes referred to as a left preconditioner.

The present paper describes two new adaptive methods for determining preconditioners during the iterations with the restarted GMRES(m) algorithm. The standard implementation of the restarted GMRES(m) algorithm [22] is based on the Arnoldi process [1], described in Section 2, and this allows spectral information of A to be gathered during the iterations. We use this information to determine an approximation of an invariant subspace of A associated with eigenvalues close to the origin. Our preconditioner essentially removes the influence of these eigenvalues on the rate of convergence. We will focus on the effect of the preconditioner on the spectrum of A , however, it is known that the rate of convergence of the iterates computed by the GMRES(m) algorithm also is determined by pseudospectra of A ; see Nachtigal et al. [19]. For ease of presentation, we ignore the effect of the preconditioner on the pseudospectra in the present paper. Our preconditioners are particularly effective when there is a cluster of a few eigenvalues of A that have a large influence on the rate of convergence. A few illustrations can be found in Section 4. The determination as well as the application of our preconditioners does not require the evaluation of any matrix-vector products with the matrix A in addition to those needed for the Arnoldi process and for the evaluation of certain residual errors. The implementations use the recurrence formulas of the Implicitly Restarted Arnoldi (IRA) method described by Sorensen [23] and more recently by Lehoucq [17]. Our preconditioners can be combined with other preconditioners, and are also applicable when no other known efficient preconditioner is available.

A different method to adaptively determine a preconditioner during iterations by the restarted GMRES(m) algorithm has recently been described by Erhel et al. [11]. By utilizing the recurrence formulas of the IRA method, our preconditioning scheme allows more flexibility in the choice of preconditioner and requires less computer memory than the method described in [11]. Another adaptive preconditioning method has been presented by Kharchenko and Yeremin [16]. Their method differs from our schemes in how approximate invariant subspaces are determined. Morgan [18] also uses approximate invariant subspaces to improve the rate of convergence of the restarted GMRES(m) algorithm; instead of constructing a preconditioner, he appends an approximate invariant subspace to the Krylov subspaces generated by the Arnoldi process. We feel that our new algorithms are attractive because of their simplicity, and because the IRA method, on which our algorithms are based, typically determines adequate approximate invariant subspaces fairly rapidly.

Let the matrix A have the spectral factorization

$$(1.6) \quad A = SAS^{-1}, \quad \Lambda = \text{diag}[\lambda_1, \lambda_2, \dots, \lambda_n].$$

Then (1.3) and (1.4) yield the bound

$$(1.7) \quad \|r_m\| \leq \|S\| \|S^{-1}\| \|r_0\| \min_{p \in \Pi_n^0} \left(\max_{z \in \lambda(A)} |p(z)| \right),$$

where $\lambda(A)$ denotes the spectrum of A . Note that the bound (1.7) would decrease if we were able to replace $\lambda(A)$ by a subset. Our preconditioners have roughly this effect. For definiteness, assume that the eigenvalues of A have been ordered according to

$$(1.8) \quad 0 < |\lambda_1| \leq |\lambda_2| \leq \dots \leq |\lambda_n|,$$

and let A be scaled so that

$$(1.9) \quad |\lambda_n| = 1.$$

A good approximation of such a scaling of A can be determined during the iterations. This is discussed below.

The Arnoldi process determines a decomposition of the form

$$(1.10) \quad AV_m = V_m H_m + f_m e_m^T,$$

where $V_m \in \mathbf{R}^{n \times m}$, $f_m \in \mathbf{R}^n$, $V_m^T V_m = I_m$, $V_m^T f_m = 0$, and $H_m \in \mathbf{R}^{m \times m}$ is an upper Hessenberg matrix. We refer to (1.10) as an Arnoldi decomposition of A . Throughout this paper e_j denotes the j th axis vector of appropriate dimension, and I_j denotes the identity matrix of order j . When $V_m e_1 = r_0 / \|r_0\|$, the columns of V_m span the Krylov subspace $\mathbf{K}_m(A, r_0)$ defined by (1.2). For future reference, we define

$$(1.11) \quad \gamma_m = \|f_m\|.$$

Let the matrix $V_k \in \mathbf{R}^{n \times k}$ consist of the first k columns of V_m , and let the columns of the matrix W_{n-k} span $\mathbf{R}^n \setminus \text{span}\{V_k\}$, where $\text{span}\{V_k\}$ denotes the span of the columns of V_k . Assume that $W_{n-k}^T W_{n-k} = I_{n-k}$. Thus, the columns of the matrix $[V_k \ W_{n-k}]$ form an orthogonal basis of \mathbf{R}^n . Introduce the matrix

$$(1.12) \quad M = V_k H_k V_k^T + W_{n-k} W_{n-k}^T.$$

We will use the inverse of matrices of the form (1.12) with $k \ll n$ as left preconditioners. The form of the inverse is given below.

PROPOSITION 1.1. *Let $Q \in \mathbf{R}^{n \times n}$ be an orthogonal matrix, and partition it according to $Q = [V \ W]$, where the submatrix V consists of the k first columns of Q , and the submatrix W consists of the remaining columns. Assume that $H = V^T A V$ is nonsingular. Then the matrix*

$$(1.13) \quad M = V H V^T + W W^T$$

is nonsingular, and its inverse is given by

$$(1.14) \quad M^{-1} = V H^{-1} V^T + W W^T.$$

Proof. The matrix (1.13) can be written as

$$M = \begin{bmatrix} V & W \end{bmatrix} \begin{bmatrix} H & 0 \\ 0 & I_{n-k} \end{bmatrix} \begin{bmatrix} V^T \\ W^T \end{bmatrix},$$

and therefore

$$(1.15) \quad M^{-1} = \begin{bmatrix} V & W \end{bmatrix} \begin{bmatrix} H^{-1} & 0 \\ 0 & I_{n-k} \end{bmatrix} \begin{bmatrix} V^T \\ W^T \end{bmatrix}.$$

This shows (1.14). \square

When the columns of the matrix V in Proposition 1.1 span an invariant subspace of A , the eigenvalues of the matrix $M^{-1}A$ can be expressed in terms of the eigenvalues of A .

COROLLARY 1.2. *Let the matrices V , W and H be as in Proposition 1.1, and assume, moreover, that the columns of the matrix V span an invariant subspace of A associated with the eigenvalues $\lambda_1, \lambda_2, \dots, \lambda_k$. Then*

$$\lambda(M^{-1}A) = \{\lambda_{k+1}, \lambda_{k+2}, \dots, \lambda_n, 1, 1, \dots, 1\},$$

where the eigenvalue 1 has multiplicity at least k .

Proof. The matrix A is similar to

$$(1.16) \quad \tilde{A} = \begin{bmatrix} V^T \\ W^T \end{bmatrix} A \begin{bmatrix} V & W \end{bmatrix} = \begin{bmatrix} H & \tilde{A}_{12} \\ 0 & \tilde{A}_{22} \end{bmatrix},$$

and $\lambda(\tilde{A}_{22}) = \{\lambda_{k+1}, \lambda_{k+2}, \dots, \lambda_n\}$. Formula (1.16) and the representation (1.15) yield

$$M^{-1}A = \begin{bmatrix} V & W \end{bmatrix} \begin{bmatrix} I_k & H^{-1}\tilde{A}_{12} \\ 0 & \tilde{A}_{22} \end{bmatrix} \begin{bmatrix} V^T \\ W^T \end{bmatrix}.$$

Thus, the spectrum of $M^{-1}A$ consists of $\lambda(\tilde{A}_{22})$ and the eigenvalue 1. The multiplicity of the latter eigenvalue is at least k . \square

A result analogous to Corollary 1.2 for a right preconditioner is shown by Erhel et al. [11]. We remark that application of preconditioners of the form (1.14) is simplified by the fact that

$$(1.17) \quad WW^T = I_n - VV^T.$$

Thus, the matrix W does not have to be computed.

The following example compares bounds for the rate of convergence of iterates determined by the GMRES(m) algorithm when applied to the original linear system (1.1) and to the preconditioned linear system (1.5) with the preconditioner (1.14), where we assume that the conditions of Corollary 1.2 hold.

Example 1.1. Assume that A has a spectral factorization of the form (1.6) with all eigenvalues real and positive, and let the eigenvalues be ordered according to (1.8). Then (1.7) yields that

$$(1.18) \quad \limsup_{m \rightarrow \infty} \left(\frac{\|r_m\|}{\|r_0\|} \right)^{1/m} \leq \limsup_{m \rightarrow \infty} \min_{p \in \Pi_m^0} \left(\max_{\lambda_1 \leq z \leq \lambda_n} |p(z)| \right)^{1/m}$$

$$(1.19) \quad = \limsup_{m \rightarrow \infty} \left(\max_{\lambda_1 \leq z \leq \lambda_n} \left| \frac{T_m\left(\frac{-2z + \lambda_n + \lambda_1}{\lambda_n - \lambda_1}\right)}{T_m\left(\frac{\lambda_n + \lambda_1}{\lambda_n - \lambda_1}\right)} \right| \right)^{1/m}$$

$$= \frac{\kappa^{1/2} - 1}{\kappa^{1/2} + 1}, \quad \kappa = \lambda_n / \lambda_1,$$

where $T_m(z)$ is the Chebyshev polynomial of the first kind of degree m for the interval $[-1, 1]$, i.e., $T_m(z) = \cos(m \arccos(z))$ for $-1 \leq z \leq 1$, and the equality (1.19) follows from well-known properties of the Chebyshev polynomials; see, e.g., [13, Section 10.1.5].

Let M^{-1} be the preconditioner (1.14), and assume that the conditions of Corollary 1.2 hold. This preconditioner eliminates the influence of the k smallest eigenvalues of A on the rate of convergence of the GMRES(m) algorithm. Specifically, the GMRES(m) algorithm applied to the preconditioned linear system (1.5) yields a sequence of residual vectors that can be bounded by

$$(1.20) \quad \limsup_{m \rightarrow \infty} \left(\frac{\|r_m\|}{\|r_0\|} \right)^{1/m} \leq \frac{\tilde{\kappa}^{1/2} - 1}{\tilde{\kappa}^{1/2} + 1} \leq \frac{\kappa^{1/2} - 1}{\kappa^{1/2} + 1}, \quad \tilde{\kappa} = \lambda_n / \lambda_{k+1},$$

where, as usual, $r_m = b - Ax_m$. The bound (1.20) can be shown by first noting that

$$(1.21) \quad \limsup_{m \rightarrow \infty} \left(\frac{\|r_m\|}{\|r_0\|} \right)^{1/m} \leq \limsup_{m \rightarrow \infty} \left(\frac{\|M^{-1}r_m\|}{\|M^{-1}r_0\|} \right)^{1/m},$$

and then applying the bound (1.18) to the right-hand side of (1.21). \square

In actual computations, we determine a preconditioner from a Krylov subspace $\text{span}\{V_k\}$, which is close to an invariant subspace. The computations of Example 1.1 suggest that the GMRES(m) algorithm will require fewer iterations to determine an accurate approximate solution of (1.1) when applied to the preconditioned linear system (1.5) with such a preconditioner than when applied to the original unpreconditioned system (1.1). This is verified by numerical experiments, some of which are presented in Section 4.

2. Construction of the preconditioners. In this section we describe how to determine an approximate invariant subspace of A associated with the eigenvalues $\lambda_1, \lambda_2, \dots, \lambda_k$, by using the recursion formulas of the IRA method of Sorensen [23]. We first present the Arnoldi process [1].

ALGORITHM 2.1. ARNOLDI PROCESS

Input: k, m , upper Hessenberg matrix $H_k = [h_{j\ell}]_{j,\ell=1}^k \in \mathbf{R}^{k \times k}$, $V_k \in \mathbf{R}^{n \times k}$, $f_k \in \mathbf{R}^n \setminus \{0\}$, such that $H_k = V_k^T A V_k$, $V_k^T V_k = I_k$, $V_k^T f_k = 0$;
Output: upper Hessenberg matrix $H_m = [h_{j\ell}]_{j,\ell=1}^m \in \mathbf{R}^{m \times m}$, $V_m \in \mathbf{R}^{n \times m}$, $f_m \in \mathbf{R}^n$, such that $H_m = V_m^T A V_m$, $V_m^T V_m = I_m$, $V_m^T f_m = 0$;
 $\gamma_k := \|f_k\|$;
if $k > 0$ *then* $h_{k+1,k} := \gamma_k$; *endif*
for $j = k + 1, k + 2, \dots, m$ *do*
 $v_j := f_{j-1} / \gamma_{j-1}$;
 for $\ell = 1, 2, \dots, j$ *do* $h_{\ell j} := v_\ell^T A v_j$; *endfor* ℓ ;
 $f_j := A v_j - \sum_{i=1}^j h_{ij} v_i$;
 $\gamma_j := \|f_j\|$; $h_{j+1,j} := \gamma_j$;
endfor j ;

\square

We may assume that all vectors f_j , $k \leq j < m$, generated by Algorithm 2.1 are nonvanishing, because if $f_j = 0$, then the columns of the matrix V_j generated span an invariant subspace of A , and V_j can be used to construct a preconditioner as described in Example 1.1. When $k = 0$ on input to Algorithm 2.1, only the initial vector f_0 has

to be provided. We note that if $f_m \neq 0$, then we can define the matrix

$$(2.1) \quad V_{m+1} = [V_m \ f_m/\gamma_m] \in \mathbf{R}^{n \times (m+1)}$$

with orthogonal columns. In the sequel, we will use the matrix

$$(2.2) \quad \tilde{H}_m = V_{m+1}^T A V_m.$$

This is an $(m+1) \times m$ matrix of Hessenberg-type, whose leading $m \times m$ principal submatrix is H_m , and whose $(m+1)$ st row is $\gamma_m e_m^T$.

Numerical difficulties can arise in the computation of the vectors f_j in the algorithm. The computations are done by the modified Gram-Schmidt process, with one reorthogonalization. Neglecting to enforce orthogonality of each vector f_j against the vectors v_1, v_2, \dots, v_j can give rise to spurious eigenvalues of the matrix H_m , i.e., eigenvalues which cannot be considered approximations of eigenvalues of A .

Given the Arnoldi decomposition (1.10) with initial vector $v_1 = V_m e_1$, the recursion formulas of the IRA method can be used to compute the vector

$$(2.3) \quad v_1^{(m-k)} = \eta_{m-k} \psi_{m-k}(A) v_1,$$

for any monic polynomial

$$\psi_{m-k}(z) = \prod_{j=1}^{m-k} (z - z_j)$$

of degree $m-k$ without evaluating any new matrix-vector products with the matrix A . The coefficient η_{m-k} is a scaling factor chosen so that $\|v_1^{(m-k)}\| = 1$. We will discuss the selection of the zeros z_j below.

The recursion formulas of the IRA method are truncated versions of the recursion formulas for the QR algorithm with explicit shifts, with the zeros z_j chosen as shifts; see, e.g., [13, Chapter 7] for a description of the QR algorithm. We therefore sometimes refer to the zeros z_j as shifts. Thus, let the decomposition (1.10) be given, and determine the QR factorization $H_m - z_1 I_m = Q^{(1)} R^{(1)}$, where $z_1 \in \mathbf{C}$, $Q^{(1)}, R^{(1)} \in \mathbf{R}^{m \times m}$, $(Q^{(1)})^T Q^{(1)} = I_m$ and $R^{(1)}$ is upper triangular. Putting $V = V_m$, $H = H_m$ and $I = I_m$, we obtain

$$(2.4.1) \quad (A - z_1 I)V - V(H - z_1 I) = f_m e_m^T,$$

$$(2.4.2) \quad (A - z_1 I)V - VQ^{(1)}R^{(1)} = f_m e_m^T,$$

$$(2.4.3) \quad (A - z_1 I)(VQ^{(1)}) - (VQ^{(1)})(R^{(1)}Q^{(1)}) = f_m e_m^T Q^{(1)},$$

$$(2.4.4) \quad A(VQ^{(1)}) - (VQ^{(1)})(R^{(1)}Q^{(1)} + z_1 I) = f_m e_m^T Q^{(1)}.$$

Let $V^{(1)} = VQ^{(1)}$ and $H^{(1)} = R^{(1)}Q^{(1)} + z_1 I$. Then $H^{(1)}$ is also a Hessenberg matrix. Multiplication of equation (2.4.2) by e_1 yields

$$(2.5) \quad (A - z_1 I)v_1 = v_1^{(1)} \rho_{11}^{(1)},$$

where $\rho_{11}^{(1)} = e_1^T R^{(1)} e_1$ and $v_1^{(1)} = V^{(1)} e_1$. Equation (2.5) displays the relationship between the initial Arnoldi vector v_1 and the vector $v_1^{(1)}$. After applying $m-k$ shifts z_1, z_2, \dots, z_{m-k} , we obtain the decomposition

$$(2.6) \quad AV_m^{(m-k)} = V_m^{(m-k)} H_m^{(m-k)} + f_m e_m^T Q,$$

where

$$V_m^{(m-k)} = [v_1^{(m-k)}, v_2^{(m-k)}, \dots, v_m^{(m-k)}] = V_m Q, \quad H_m^{(m-k)} = Q^T H_m Q.$$

Here $Q = Q^{(1)}Q^{(2)} \dots Q^{(m-k)}$, and $Q^{(j)}$ denotes the orthogonal matrix associated with the shift z_j . Introduce the partitioning

$$H_m^{(m-k)} = \begin{bmatrix} H_k^{(m-k)} & G \\ \tilde{\gamma}_k e_1 e_k^T & \hat{H}_{m-k} \end{bmatrix},$$

and equate the first k columns on the right-hand side and left-hand side of (2.6). This gives

$$(2.7) \quad AV_k^{(m-k)} = V_k^{(m-k)} H_k^{(m-k)} + f_k^{(m-k)} e_k^T,$$

where

$$(2.8) \quad f_k^{(m-k)} = V_{m-k}^{(m-k)} e_1 \tilde{\gamma}_k + f_m e_m^T Q e_k$$

and $V_m^{(m-k)} = [V_k^{(m-k)} \ V_{m-k}^{(m-k)}]$. It follows from

$$(V_k^{(m-k)})^T V_{m-k}^{(m-k)} e_1 = 0, \quad (V_k^{(m-k)})^T f_m = 0$$

and (2.8), that $(V_k^{(m-k)})^T f_k^{(m-k)} = 0$. Thus, (2.7) is an Arnoldi decomposition of A . By construction, the vector $v_1^{(m-k)} = V_k^{(m-k)} e_1$ can be written as (2.3).

While our description of the IRA method is based on recursion formulas for the QR algorithm with explicit shifts, our implementation is based on the QR algorithm with implicit shifts for reason of numerical stability; see [13, Chapter 7] for a description of this QR algorithm. The use of implicit shifts allows the application of complex conjugate shifts without using complex arithmetic.

We first apply the Arnoldi process to compute the Arnoldi decomposition (1.10), and then use the recursion formulas of the IRA method to determine the Arnoldi decomposition (2.7). The purpose of these computations is to determine an accurate approximation of an invariant subspace of A associated with the eigenvalues $\lambda_1, \lambda_2, \dots, \lambda_k$. We would like to choose the zeros z_1, z_2, \dots, z_{m-k} of ψ_{m-k} , so that the first column $v_1^{(m-k)}$ of $V_k^{(m-k)}$ defined by (2.3) is in, or close to, an invariant subspace of A associated with the eigenvalues $\lambda_1, \lambda_2, \dots, \lambda_k$.

Let $\{\theta_j^{(m)}\}_{j=1}^m$ denote the eigenvalues of the upper Hessenberg matrix H_m in (1.10), and order them so that

$$(2.9) \quad |\theta_1^{(m)}| \leq |\theta_2^{(m)}| \leq \dots \leq |\theta_m^{(m)}|.$$

Since $H_m = V_m^T A V_m$ is an orthogonal projection of A , we consider the $\theta_j^{(m)}$ to be approximations of eigenvalues of A . In order to force the vector $v_1^{(m-k)}$ into an invariant subspace of A associated with the k eigenvalues of A of smallest magnitude, we choose the zeros

$$(2.10) \quad z_j = \theta_{m+1-j}^{(m)}, \quad 1 \leq j \leq m-k,$$

i.e., the z_j are chosen to be available approximations of the $m-k$ eigenvalues of A of largest magnitude. This selection of zeros is discussed by Sorensen [23], Calvetti

et al. [6] and Lehoucq [17], and these zeros are there referred to as “exact shifts”. Numerical experience indicates that the ordering of the shifts according to (2.10) is adequate in the sense that the computed matrices $H_k^{(m-k)} \in \mathbf{R}^{k \times k}$ have eigenvalues very close to the k eigenvalues of H_m of smallest magnitude.

Let $\{\theta_j^{(k)}, y_j^{(k)}\}_{j=1}^k$ be eigenvalue-eigenvector pairs of $H_k^{(m-k)}$, and introduce the vectors $x_j = V_k^{(m-k)} y_j^{(k)}$, $1 \leq j \leq k$. Then $\{\theta_j^{(k)}, x_j\}_{j=1}^k$ are approximate eigenvalue-eigenvector pairs of A with residual errors

$$\|Ax_j - x_j \theta_j^{(k)}\| = \|(AV_k^{(m-k)} - V_k^{(m-k)} H_k^{(m-k)}) y_j^{(k)}\| = \|f_k^{(m-k)}\| |e_k^T y_j^{(k)}|, \quad 1 \leq j \leq k.$$

We accept $\text{span}\{V_k\}$ as an approximate invariant subspace of A if

$$(2.11) \quad \|f_k^{(m-k)}\| |e_k^T y_j^{(k)}| \leq \|H_k^{(m-k)}\| \epsilon_{\text{subspace}}, \quad 1 \leq j \leq k,$$

where $\epsilon_{\text{subspace}} > 0$ is a parameter. The purpose of the matrix $H_k^{(m-k)}$ in (2.11) is to make the bound invariant under scaling of A .

If the inequalities (2.11) are not satisfied, then we apply Algorithm 2.1 with the Arnoldi decomposition (2.7) as input in order to determine a new Arnoldi decomposition (1.10) with an $m \times m$ upper Hessenberg matrix H_m . We then again apply the recursion formulas of the IRA method with the zeros chosen to be the $m - k$ eigenvalues of H_m of largest magnitude. This gives an Arnoldi decomposition of the form (2.7), and we check whether the inequalities (2.11) are satisfied. The computations are repeated in this fashion until (2.11) holds. We obtain in this way an Arnoldi decomposition of the form (2.7) with matrices $V_k = V_k^{(m-k)}$ and $H_k = H_k^{(m-k)}$, such that, in general, $\text{span}\{V_k\}$ is an accurate approximation of an invariant subspace associated with the k eigenvalues of smallest magnitude of A , and $\lambda(H_k)$ is an accurate approximation of the set $\{\lambda_j\}_{j=1}^k$. The accuracy of the approximations depends on the parameter $\epsilon_{\text{subspace}}$ in (2.11), the distribution of the eigenvalues of A , and the departure from normality of A . The matrices V_k and H_k so obtained are used to define our first preconditioner

$$(2.12) \quad M_1^{-1} = V_k H_k^{-1} V_k^T + I - V_k V_k^T,$$

where we have used (1.17).

We describe in Section 3 how to combine the IRA process with the restarted GMRES algorithm and Richardson iteration, so that we can improve an available approximate solution of (1.1) while determining the preconditioner M_1^{-1} .

Having computed the preconditioner $M^{-1} = M_1^{-1}$, we apply the method outlined above to the preconditioned system (1.5) in order to determine an approximation of an invariant subspace associated with the eigenvalues of smallest magnitude of the matrix $M_1^{-1}A$, and simultaneously improve an available approximate solution of (1.1). This yields a new preconditioner M_2^{-1} for the system $M_1^{-1}Ax = M_1^{-1}b$, or equivalently, a new preconditioner $M^{-1} = M_2^{-1}M_1^{-1}$ for the system (1.1). The computations are continued in this manner until we have determined a preconditioner of the form

$$(2.13) \quad M^{-1} = M_{\alpha_0}^{-1} M_{\alpha_0-1}^{-1} \cdots M_1^{-1},$$

for some specified integer $\alpha_0 \geq 1$. The form (2.13) of the preconditioner makes it natural to scale A so that (1.9) holds. An approximation of such a scaling is achieved by scaling the linear system (1.1) by the factor $1/|\theta_m^{(m)}|$, where $\theta_m^{(m)}$ is an eigenvalue

of largest magnitude of one of the matrices H_m generated by Algorithm 2.1 during our computation of the preconditioner M_1^{-1} . We remark that for certain matrices A other techniques for achieving such a scaling may be available. For instance, one may be able to use Gershgorin disks or the inequality $|\lambda_n| \leq \|A\|_*$, where $\|\cdot\|_*$ is any matrix norm induced by a vector norm; see [24, Chapter 6] for details on the latter topics.

3. The iterative methods. This section describes our two algorithms for adaptive preconditioning in detail. One of them, Algorithm 3.5, combines the IRA process with Richardson iteration and the GMRES algorithm. The other scheme, Algorithm 3.6, does not apply Richardson iteration. We first recall the restarted GMRES(m) algorithm by Saad and Schultz [22] for the solution of linear systems of equations (1.1).

ALGORITHM 3.1. RESTARTED GMRES(m) ALGORITHM

Input: m , initial approximate solution $x_0 \in \mathbf{R}^n$, $\epsilon_{\text{solution}} \geq 0$.

Output: approximate solution x_m , associated residual vector r_m .

$r_m := r_0 := b - Ax_0$; $x_m := x_0$;

while $\|r_m\|/\|r_0\| > \epsilon_{\text{solution}}$ *do*

Compute Arnoldi decomposition (1.10) by Algorithm 2.1 with input $k = 0$ and $f_0 = r_m$. Then the matrices V_{m+1} and \bar{H}_m , defined by (2.1) and (2.2), respectively, are also available.

Compute solution $y_m \in \mathbf{R}^m$ of $\min_{y \in \mathbf{R}^m} \|\|r_m\|e_1 - \bar{H}_m y\|$.

$x_m := x_m + V_m y_m$; $r_m := r_m - V_{m+1} \bar{H}_m y_m$;

endwhile;

□

We now describe how to improve an available approximate solution by Richardson iteration while applying the recursion formulas of the IRA method to an Arnoldi decomposition. These iterations can be carried out without evaluating matrix-vector products with the matrix A . Let x_0 be an available approximate solution of (1.1). Richardson iteration can be written as

$$(3.1) \quad x_j = x_{j-1} + \delta_j r_{j-1}, \quad j = 1, 2, \dots,$$

$$(3.2) \quad r_{j-1} = b - Ax_{j-1},$$

where the $\delta_j \in \mathbf{C}$ are relaxation parameters. We would like the parameters δ_j to be such that the approximate solutions x_j converge rapidly to the solution of (1.1) as j increases. For future reference, we note that the residual vectors (3.2) can be written as

$$(3.3) \quad r_{j-1} = \prod_{\ell=1}^{j-1} (I - \delta_\ell A) r_0.$$

THEOREM 3.2. *Let x_0 be an approximate solution of (1.1), and let $r_0 = b - Ax_0$. Consider the Arnoldi decomposition $AV_m = V_m H_m + f_m e_m^T$ with the initial vector $v_1 = V_m e_1$ given by $v_1 = r_0/\|r_0\|$. Apply the recursion formulas of the IRA method with zeros z_1, z_2, \dots, z_j for some $j \leq m$. Then the residual vectors (3.2) associated with the iterates (3.1) computed by Richardson iteration with relaxation parameters*

$\delta_j = 1/z_j$ are given by

$$(3.4) \quad r_j = (-1)^j \|r_0\| \prod_{\ell=1}^j \delta_\ell \rho_{11}^{(\ell)} v_1^{(j)}, \quad 1 \leq j \leq m,$$

where $v_1^{(j)} = V_m Q^{(1)} Q^{(2)} \dots Q^{(j)} e_1$ and $\rho_{11}^{(j)} = e_1^T R^{(j)} e_1$ for $j < m$. Here $Q^{(\ell)}$ denotes the orthogonal matrix and $R^{(\ell)}$ the upper triangular matrix associated with the zero z_ℓ in the IRA recursion formulas. Moreover, $v_1^{(m)} = (h_{11}^{(m-1)} - z_m) v_1^{(m-1)} + f_1^{(m-1)}$ and $\rho_{11}^{(m)} = 1$.

Proof. We first show (3.4) for $j = 1$. Substitution of $v_1 = r_0 / \|r_0\|$ into (2.5) yields

$$(A - z_1 I) r_0 = \|r_0\| v_1^{(1)} \rho_{11}^{(1)}.$$

The representation (3.3) now shows that

$$(3.5) \quad r_1 = (I - \delta_1 A) r_0 = -\delta_1 (A - z_1 I) r_0 = \|r_0\| (-\delta_1) v_1^{(1)} \rho_{11}^{(1)}.$$

We turn to the case when $j = 2$. From (3.3) and (3.5), we obtain

$$(3.6) \quad r_2 = \delta_1 \delta_2 (A - z_2 I) (A - z_1 I) r_0 = \|r_0\| \delta_1 \delta_2 \rho_{11}^{(1)} (A - z_2 I) v_1^{(1)}.$$

Replace $V = V_m$ by $V_m Q^{(1)}$ in equations (2.4.1)-(2.4.4), and multiply the equation (2.4.2) so obtained by e_1 . This shows, analogously to (2.5), that

$$(3.7) \quad (A - z_2 I) v_1^{(1)} = v_1^{(2)} \rho_{11}^{(2)}.$$

Substitution of (3.7) into (3.6) shows (3.4) for $j = 2$. Continuing in this manner yields (3.4) for all $j < m$.

The case $j = m$ has to be treated separately. We have the Arnoldi decomposition

$$(A - z_m I) v_1^{(m-1)} = (h_{11}^{(m-1)} - z_m) v_1^{(m-1)} + f_1^{(m-1)},$$

and similarly as in [3], we obtain $v_1^{(m)} = (h_{11}^{(m-1)} - z_m) v_1^{(m-1)} + f_1^{(m-1)}$. Choosing $\rho_{11}^{(m)} = 1$ completes the proof. \square

Prior to the development of the GMRES(m) algorithm, Saad [20] introduced the Full Orthogonalization (FO(m)) algorithm. This is a Galerkin method for the solution of (1.1). Let x_0 be an approximate solution of (1.1) and let r_0 be the associated residual vector. Consider the Arnoldi decomposition (1.10), and let $v_1 = V_m e_1$ be the same as in Theorem 3.2. The FO(m) algorithm determines an improved approximate solution x_m of (1.1) by solving the linear system

$$(3.8) \quad V_m^T A V_m y_m = V_m^T r_0$$

and then letting

$$(3.9) \quad x_m = x_0 + V_m y_m.$$

The following result shows how this approximate solution can be determined by Richardson iteration.

THEOREM 3.3. *Let the vectors x_0 and r_0 , and the Arnoldi decomposition (1.10), be the same as in Theorem 3.2. Assume that the Arnoldi decomposition exists with*

$\|f_m\| \neq 0$ and that the matrix H_m in the Arnoldi decomposition is nonsingular. Let $j = m$ in (3.4), and let the relaxation parameters for Richardson iteration be reciprocal values of the eigenvalues of H_m . Then, in exact arithmetic, the approximate solution x_m determined by Richardson iteration (3.1)-(3.2) equals the approximate solution (3.9) computed by the FO(m) algorithm.

Proof. Substitute (3.9) into $r_m = b - Ax_m$, and use the fact that the linear system (3.8) can be written as $H_m y_m = e_1 \|r_0\|$, to obtain

$$(3.10) \quad r_m = -f_m e_m^T y_m.$$

Introduce, for polynomials f and g , the bilinear form

$$\langle f, g \rangle = r_0^T f(A^T) g(A) r_0.$$

By construction,

$$(3.11) \quad \begin{aligned} v_{j+1} &= V_m e_{j+1} = g_j(A) r_0, & 0 \leq j < m, \\ f_m &= \|f_m\| g_m(A) r_0, \end{aligned}$$

where g_j is a polynomial of degree j . The g_j , $0 \leq j \leq m$, satisfy

$$\langle g_j, g_\ell \rangle = \begin{cases} 1, & \text{if } j = \ell, \\ 0, & \text{if } j \neq \ell. \end{cases}$$

In particular, equations (3.10) and (3.11) yield

$$r_m = -\|f_m\| e_m^T y_m g_m(A) r_0,$$

which shows that $p_m(t) = -\|f_m\| e_m^T y_m g_m(t)$ is the residual polynomial of degree m for the FO(m) algorithm, and therefore satisfies $p_m(0) = 1$. Combining formulas (1.10) and (3.11) yields the identity

$$t[g_0(t), g_1(t), \dots, g_{m-1}(t)] = [g_0(t), g_1(t), \dots, g_{m-1}(t)] H_m + \|f_m\| g_m(t) e_m^T,$$

which shows that the eigenvalues $\{\theta_j^{(m)}\}_{j=1}^m$ of H_m are the zeros of g_m . In particular, all $\theta_j^{(m)} \neq 0$, and therefore p_m can be written as $p_m(t) = g_m(t)/g_m(0)$. It follows that

$$(3.12) \quad p_m(t) = \prod_{j=1}^m (1 - t/\theta_j^{(m)}).$$

A comparison of (3.12) with (1.3) and (3.3) shows that m steps of Richardson iteration with relaxation parameters $\delta_j = 1/\theta_j^{(m)}$ and an application of the FO(m) algorithm correspond to the same residual polynomial, and therefore are equivalent. \square

The implementation of our iterative method is based on the following observation.

COROLLARY 3.4. *Let x_{j-1} be an approximate solution of (1.1), and let $r_{j-1} = b - Ax_{j-1}$ be the associated residual vector. Let*

$$(3.13) \quad AV_\ell = V_\ell H_\ell + f_\ell e_\ell^T, \quad \ell > 1,$$

be an Arnoldi decomposition, with initial vector $v_1 = V_\ell e_1 = r_{j-1}/\|r_{j-1}\|$. Denote by $\{\theta_j^{(\ell)}\}_{j=1}^\ell$ the eigenvalues of H_ℓ , let $x_j = x_{j-1} - \delta_j r_{j-1}$ be the approximate solution

obtained by one step of Richardson iteration with relaxation parameter $\delta_j = 1/\theta_q^{(\ell)}$, for some $1 \leq q \leq \ell$, and let an application of the recursion formulas of the IRA method to (3.13) with shift $\theta_q^{(\ell)}$ yield the Arnoldi decomposition

$$(3.14) \quad AV_{\ell-1}^{(1)} = V_{\ell-1}^{(1)}H_{\ell-1}^{(1)} + f_{\ell-1}^{(1)}e_{\ell-1}^T.$$

Then $r_j = -\|r_{j-1}\|\delta_j\rho_{11}^{(1)}v_1^{(1)}$, where $v_1^{(1)} = V_{\ell-1}^{(1)}e_1$ and $\rho_{11}^{(1)} = e_1^T R^{(1)}e_1$. Here $R^{(1)}$ is the triangular matrix in a QR factorization of $H_\ell - \theta_q^{(\ell)}I_\ell$. Moreover, $\lambda(H_{\ell-1}^{(1)}) = \{\theta_j^{(\ell)}\}_{j=1}^{q-1} \cup \{\theta_j^{(\ell)}\}_{j=q+1}^\ell$.

Proof. The Corollary follows from Theorem 3.2 and the fact that when we use an exact shift, the eigenvalues of the reduced matrix $H_{\ell-1}^{(1)}$ are the eigenvalues of the original matrix H_ℓ , except for the shift. The latter result is shown by Sorensen [23, Lemma 3.10]. \square

The corollary above shows that we can apply $m - k$ shifts, one at a time, and determine the required residual vectors from the first column of the matrices V_ℓ in the available Arnoldi decompositions. An analogous result can be established for complex conjugate shifts. In the latter case, the recursion formulas for the IRA method are implemented by using the QR algorithm with implicit double shifts. This obviates the need to use complex arithmetic. A double step of Richardson iteration, with complex conjugate relaxation parameters, also can be carried out without using complex arithmetic. For notational simplicity, the algorithm below for our iterative method does not use double shifts and double steps, however, our implementation of the algorithm used for the computed examples of Section 4 does.

ALGORITHM 3.5. ADAPTIVELY PRECONDITIONED GMRES(m) ALGORITHM
WITH RICHARDSON ITERATION

Input: tolerance for computed approximate solution $\epsilon_{\text{solution}}$, tolerance for approximate invariant subspace $\epsilon_{\text{subspace}}$, dimension m of largest Krylov subspace determined, dimension k of approximate invariant subspace to be computed, maximum number α_0 of preconditioners M_j^{-1} to be computed, maximum number β_0 of Arnoldi decompositions of order m to be determined for each preconditioner M_j^{-1} .

Output: computed approximate solution x_j , associated residual vector r_j , preconditioner $M^{-1} = M_\alpha^{-1}M_{\alpha-1}^{-1} \cdots M_1^{-1}$ for some $\alpha \leq \alpha_0$.

$M^{-1} := I$; $x_0 := 0$; $r_0 := b$; $j := 0$;

for $\alpha = 1, 2, \dots, \alpha_0$ do

Compute Arnoldi decomposition $M^{-1}AV_m = V_mH_m + f_m e_m^T$ by Algorithm 2.1 with initial vector $v_1 = M^{-1}r_0/\|M^{-1}r_0\|$.

for $\beta = 1, 2, \dots, \beta_0$ do

Compute eigenvalues $\{\theta_j^{(m)}\}_{j=1}^m$ of matrix H_m in Arnoldi decomposition and order them according to (2.9).

if $j = 0$ then scale matrix and right-hand side of linear system by factor $1/|\theta_m^{(m)}|$. Then equation (1.9) is approximately satisfied.

for $\ell = 1, 2, \dots, m - k$ do

$j := j + 1$; $\delta_j := 1/\theta_{m+1-\ell}^{(m)}$; $x_j := x_{j-1} + \delta_j M^{-1}r_{j-1}$;

Apply shift $\theta_{m+1-\ell}^{(m)}$ to Arnoldi decomposition and compute residual vector $M^{-1}r_j$ as described by Corollary 3.4. This gives

Arnoldi decomposition $M^{-1}AV_{m-\ell}^{(\ell)} = V_{m-\ell}^{(\ell)}H_{m-\ell}^{(\ell)} + f_{m-\ell}^{(\ell)}e_{m-\ell}^T$.

endfor ℓ ;
if bound (2.11) is satisfied then goto 1;
 Use the Arnoldi decomposition $M^{-1}AV_k^{(m-k)} = V_k^{(m-k)}H_k^{(m-k)} + f_k^{(m-k)}e_k^T$ as input to Algorithm 2.1 and apply the Arnoldi process to compute the Arnoldi decomposition $M^{-1}AV_m = V_mH_m + f_me_m^T$.
endfor β ;
 1: Improve approximate solution by GMRES(k) and update preconditioner:
 The Arnoldi decomposition $M^{-1}AV_k = V_kH_k + f_ke_k^T$, as well as the matrices V_{k+1} and \bar{H}_k , cf. (2.1) and (2.2), are available.
 Compute solution $y_k \in \mathbf{R}^k$ of $\min_{y \in \mathbf{R}^k} \|V_{k+1}^T M^{-1}r_j - \bar{H}_k y\|$;
 $x_{j+k} := x_j + V_k y_k$; $M^{-1}r_{j+k} := M^{-1}r_j - V_{k+1} \bar{H}_k y_k$;
 2: $M_\alpha^{-1} := V_k^{(m-k)}(H_k^{(m-k)})^{-1}(V_k^{(m-k)})^T + I - V_k^{(m-k)}(V_k^{(m-k)})^T$; $M^{-1} := M_\alpha^{-1}M^{-1}$;
 3: $r_{j+k} := b - Ax_{j+k}$; $j := j + k$;
if $\|r_j\|/\|r_0\| \leq \epsilon_{\text{solution}}$ *then done*;
endfor α ;
while $\|r_j\|/\|r_0\| > \epsilon_{\text{solution}}$ *do*
 Apply GMRES(m): compute Arnoldi decomposition $M^{-1}AV_m = V_mH_m + f_me_m^T$ by Algorithm 2.1 with initial vector $v_1 = M^{-1}r_j/\|M^{-1}r_j\|$, and determine the matrices V_{m+1} and \bar{H}_m defined by (2.1) and (2.2), respectively.
 Compute solution $y_m \in \mathbf{R}^m$ of $\min_{y \in \mathbf{R}^m} \| \|M^{-1}r_j\|e_1 - \bar{H}_m y\|$;
 $x_{j+m} := x_j + V_m y_m$; $M^{-1}r_{j+m} := M^{-1}r_j - V_{m+1} \bar{H}_m y_m$;
 4: $r_{j+m} := b - Ax_{j+m}$; $j := j + m$;
endwhile;

□

In Algorithm 3.5, we only have to compute matrix-vector products with the matrix A when applying the Arnoldi process, and when evaluating residual vectors r_ℓ in the lines labeled “3:” and “4:”.

We now examine the storage requirement of Algorithm 3.5 and count the number of n -vectors that have to be stored. Storage necessary to represent the matrix A is ignored, since it is independent of the iterative method used. Each preconditioner M_j^{-1} requires the storage of an $n \times k$ matrix V_k , and we limit the number of these preconditioners to α_0 . Thus, the preconditioner M^{-1} defined by (2.13) requires the storage of at most $\alpha_0 k$ n -vectors. In particular, the matrix M^{-1} is not actually formed. The line marked “2:” in Algorithm 3.5 is to be interpreted symbolically to mean that the storage for the matrix M^{-1} and the formula for evaluating matrix-vector products with M^{-1} are updated. The GMRES(m) algorithm in the while-loop of Algorithm 3.5 requires additional storage for the vectors x_j and r_j , and for the matrix $V_{m+1} \in \mathbf{R}^{n \times (m+1)}$. This is equivalent to the storage of $m + 3$ n -vectors. The vector $M^{-1}r_j$ in Algorithm 3.5 is up to a scaling factor stored in the first column of the matrix V_{m+1} . The last column of V_{m+1} contains the vector f_m up to a scaling factor. The right-hand side vector b also has to be stored. Therefore, the total storage requirement of Algorithm 3.5 is at most $\alpha_0 k + m + 4$ n -vectors.

Algorithm 3.6 below is obtained by replacing Richardson iteration in Algorithm 3.5 by the GMRES algorithm. This replacement makes the the residual error decrease more smoothly as the iterations proceed. However, the iterates and preconditioners generated by Algorithms 3.5 and 3.6 are not the same, and we have found that the

former algorithm not seldom gives faster convergence. This is illustrated in Section 4. We therefore feel that both algorithms are of interest. The storage requirement of Algorithm 3.6 is essentially the same as of Algorithm 3.5. For notational simplicity, Algorithm 3.6 does not use double shifts, however, our implementation of the algorithm used for the computed examples of Section 4 does.

ALGORITHM 3.6. ADAPTIVELY PRECONDITIONED GMRES(m) ALGORITHM

Input: tolerance for computed approximate solution $\epsilon_{\text{solution}}$, tolerance for approximate invariant subspace $\epsilon_{\text{subspace}}$, dimension m of largest Krylov subspace determined, dimension k of approximate invariant subspace to be computed, maximum number α_0 of preconditioners M_j^{-1} to be computed, maximum number β_0 of Arnoldi decompositions of order m to be determined for each preconditioner M_j^{-1} .

Output: computed approximate solution x_j , associated residual vector r_j , preconditioner $M^{-1} = M_\alpha^{-1}M_{\alpha-1}^{-1} \cdots M_1^{-1}$ for some $\alpha \leq \alpha_0$.

$M^{-1} := I; x_0 := 0; r_0 := b; j := 0;$

for $\alpha = 1, 2, \dots, \alpha_0$ do

Compute Arnoldi decomposition $M^{-1}AV_m = V_m H_m + f_m e_m^T$ by Algorithm 2.1 with initial vector $v_1 = M^{-1}r_0 / \|M^{-1}r_0\|$.

Apply GMRES(m): determine the matrices V_{m+1} and \bar{H}_m defined by (2.1) and (2.2), respectively.

Compute solution $y_m \in \mathbf{R}^m$ of $\min_{y \in \mathbf{R}^m} \| \|M^{-1}r_j\| e_1 - \bar{H}_m y \|$;

$x_{j+m} := x_j + V_m y_m; M^{-1}r_{j+m} := M^{-1}r_j - V_{m+1} \bar{H}_m y_m;$

$r_{j+m} := b - Ax_{j+m}; j := j+m;$

for $\beta = 1, 2, \dots, \beta_0$ do

Compute eigenvalues $\{\theta_j^{(m)}\}_{j=1}^m$ of matrix H_m in Arnoldi decomposition and order them according to (2.9).

if $j = m$ then scale matrix and right-hand side of linear system by factor $1/|\theta_m^{(m)}|$. Then equation (1.9) is approximately satisfied.

for $\ell = 1, 2, \dots, m - k$ do

$j := j + 1;$

Apply shift $\theta_{m+1-\ell}^{(m)}$ to Arnoldi decomposition by using the IRA formulas (2.4)-(2.8). This gives Arnoldi decomposition

$$M^{-1}AV_{m-\ell}^{(\ell)} = V_{m-\ell}^{(\ell)} H_{m-\ell}^{(\ell)} + f_{m-\ell}^{(\ell)} e_{m-\ell}^T.$$

endfor ℓ ;

if bound (2.11) is satisfied then goto 1;

Use the Arnoldi decomposition $M^{-1}AV_k^{(m-k)} = V_k^{(m-k)} H_k^{(m-k)} + f_k^{(m-k)} e_k^T$ as input to Algorithm 2.1 and apply the Arnoldi process to compute the Arnoldi decomposition $M^{-1}AV_m = V_m H_m + f_m e_m^T$. Apply GMRES(m): determine the matrices V_{m+1} and \bar{H}_m defined by (2.1) and (2.2), respectively.

Compute solution $y_m \in \mathbf{R}^m$ of $\min_{y \in \mathbf{R}^m} \| \|M^{-1}r_j\| e_1 - \bar{H}_m y \|$;

$x_{j+m} := x_j + V_m y_m; M^{-1}r_{j+m} := M^{-1}r_j - V_{m+1} \bar{H}_m y_m;$

$r_{j+m} := b - Ax_{j+m}; j := j+m;$

endfor β ;

1: Improve approximate solution by GMRES(k) and update preconditioner: The Arnoldi decomposition $M^{-1}AV_k = V_k H_k + f_k e_k^T$, as well as the ma-

trices V_{k+1} and \bar{H}_k , cf. (2.1) and (2.2), are available.
 Compute solution $y_k \in \mathbf{R}^k$ of $\min_{y \in \mathbf{R}^k} \|V_{k+1}^T M^{-1} r_j - \bar{H}_k y\|$;

$x_{j+k} := x_j + V_k y_k$; $M^{-1} r_{j+k} := M^{-1} r_j - V_{k+1} \bar{H}_k y_k$;

2: $M_\alpha^{-1} := V_k^{(m-k)} (H_k^{(m-k)})^{-1} (V_k^{(m-k)})^T + I - V_k^{(m-k)} (V_k^{(m-k)})^T$; $M^{-1} := M_\alpha^{-1} M^{-1}$;

3: $r_{j+k} := b - A x_{j+k}$; $j := j + k$;
 if $\|r_j\|/\|r_0\| \leq \epsilon_{\text{solution}}$ then done;

endfor α ;

while $\|r_j\|/\|r_0\| > \epsilon_{\text{solution}}$ do

 Apply GMRES(m): compute Arnoldi decomposition $M^{-1} A V_m = V_m H_m + f_m e_M^T$ by Algorithm 2.1 with initial vector $v_1 = M^{-1} r_j / \|M^{-1} r_j\|$, and determine the matrices V_{m+1} and \bar{H}_m defined by (2.1) and (2.2), respectively.
 Compute solution $y_m \in \mathbf{R}^m$ of $\min_{y \in \mathbf{R}^m} \|M^{-1} r_j \|e_1 - \bar{H}_m y\|$;

$x_{j+m} := x_j + V_m y_m$; $M^{-1} r_{j+m} := M^{-1} r_j - V_{m+1} \bar{H}_m y_m$;

4: $r_{j+m} := b - A x_{j+m}$; $j := j + m$;

endwhile;

□

The comments regarding the lines with labels “2:”, “3:” and “4:” for Algorithm 3.5 also apply to Algorithm 3.6.

4. Numerical experiments. All the numerical experiments presented in this section were carried out on an HP 9000/735 computer using MATLAB. In all examples we chose the initial approximate solution $x_0 = 0$, and this gives $r_0 = b$. The vector b had randomly generated uniformly distributed entries in the open interval $(0, 1)$. The purpose of the experiments was to compare Algorithms 3.5 and 3.6 to a restarted GMRES(m_0) algorithm, where the parameter m_0 is chosen so that the latter algorithm is allowed at least as much computer storage as the former two algorithms. We also compare Algorithms 3.5 and 3.6 to the GMRES algorithm without restarts, and refer to the latter scheme as “Full GMRES”. We terminated the iterations with these iterative methods as soon as a residual vector r_j was determined, such that

$$\frac{\|r_j\|}{\|r_0\|} \leq \epsilon_{\text{solution}},$$

with $\epsilon_{\text{solution}} = 1 \cdot 10^{-10}$. For Algorithms 3.5 and 3.6, we chose the input parameter values $\epsilon_{\text{subspace}} = 1 \cdot 10^{-4}$, $\alpha_0 = 3$, $\beta_0 = 10$, $k = 10$ and $m = 20$. The storage requirement for both Algorithms 3.5 and 3.6 with this choice of parameters is at most 54 n -vectors. We compare these schemes with the restarted GMRES(60) algorithm, which requires the storage of 62 n -vectors for V_{61} and x_m ; see Algorithm 3.1. This storage count assumes that the residual vector r_m in Algorithm 3.1 up to a scaling factor is stored in the first column of the matrix V_{61} .

Example 4.1. Let the matrix $A \in \mathbf{R}^{200 \times 200}$ be partitioned according to

$$A = \begin{bmatrix} A_{1,1} & A_{1,2} \\ A_{1,2}^T & A_{2,2} \end{bmatrix},$$

where $A_{1,1} \in \mathbf{R}^{30 \times 30}$ is a circulant matrix with first row $[-3/2, 0, \dots, 0, 2]$. The entries of the diagonal matrix $A_{2,2} \in \mathbf{R}^{170 \times 170}$ are uniformly distributed random numbers in the interval $(1, 10)$. The matrix $A_{1,2}$ is a zero matrix of appropriate order. Thus, the

the eigenvalues of A closest to the origin were removed, and Algorithms 3.5 and 3.6 yield faster convergence than the restarted GMRES(60) algorithm; see Figure 4.4. \square

Example 4.3. Let $A = A' + 1 \cdot 10^4 I$, where A' is the Pores3 matrix of the Harwell-Boeing matrix collection. The matrix A' is nonsymmetric, of order $n = 532$ and with 3474 non-zero entries. The purpose of the shift $1 \cdot 10^4$ was to obtain a matrix with some positive eigenvalues. Figures 4.5 and 4.6 are analogous to Figures 4.1 and 4.2, respectively, and Table 4.3 is analogous to Table 4.1. We can see that some eigenvalues of the matrix A are very close to the origin and others are of large magnitude. Figure 4.5 illustrates how the preconditioner moves eigenvalues of A away from the origin to approximately $\text{sign}(\text{Re}(\lambda_n))|\lambda_n|$, which is negative. Figure 4.6 shows the rate of convergence. \square

Example 4.4. Let A be a diagonal matrix of order 200 with diagonal entries

$$a_{jj} = \begin{cases} \frac{j}{100n}, & \text{for } 1 \leq j \leq 25, \\ \frac{10j}{n}, & \text{for } 26 \leq j \leq 200. \end{cases}$$

Figures 4.7 and 4.9 are analogous to Figures 4.1 and 4.2, respectively, and Table 4.4 is analogous to Table 4.1. Figure 4.8 illustrates that the preconditioner moved all the smallest eigenvalues of A , except for one, away from the origin. Figure 4.9 shows the rate of convergence. \square

Example 4.5. In all examples above, we chose the shifts according to (2.10), i.e., we determined approximations of subspaces associated with a few eigenvalues of smallest magnitude. The present example illustrates that the Algorithms 3.5 and 3.6 easily can be modified to determine approximations of other invariant subspaces. Specifically, we used Algorithm 3.6 to solve the same linear system of equations as in Example 4.1, and chose as shifts the $m - k$ eigenvalues with largest real part of the matrices H_m generated during the iterations. Thus, we sought to determine invariant subspaces associated with a few of the eigenvalues with smallest real part. Figure 4.10 is analogous to Figure 4.1 and shows $\lambda(A)$ (dots) and $\lambda(M^{-1}A)$ (stars). All 30 eigenvalues of A on the circle were removed, and the number of matrix-vector products required before the stopping criterion was satisfied was 311, which is less than the numbers of matrix-vector products reported in Table 4.1. \square

5. Conclusion. This paper describes new preconditioning methods that are well suited for use with the restarted GMRES(m) algorithm. Numerous computed examples indicate that iterates generated by our methods can converge significantly faster than iterates determined by a restarted GMRES algorithm that requires more computer storage. Algorithms 3.5 and 3.6 describe versions of our preconditioning method in which the eigenvalues λ_j of smallest magnitude of the matrix A are mapped to approximately $\text{sign}(\text{Re}(\lambda_j))|\lambda_n|$. Example 4.5 illustrates that it is easy to modify our preconditioners so that other eigenvalues are mapped.

Acknowledgements. Work on this paper was carried out while the last three authors visited the Computer Science Department and IPS at the ETH. They would like to thank Walter Gander and Martin Gutknecht for making these visits possible. We would like to thank Marcus Grote for discussions and code for extracting matrices from the Harwell-Boeing matrix collection, and Richard Lehoucq for providing us with reference [16].

REFERENCES

- [1] W. E. ARNOLDI, *The principle of minimized iterations in the solution of the matrix eigenvalue problem*, *Quart. Appl. Math.*, 9 (1951), pp. 17–29.
- [2] O. AXELSSON, *Iterative Solution Methods*, Cambridge University Press, Cambridge, 1994.
- [3] J. BAGLAMA, D. CALVETTI AND L. REICHEL, *Iterative methods for computing a few eigenvalues of a large symmetric matrix*, *BIT*, to appear.
- [4] Z. BAI, D. HU AND L. REICHEL, *A Newton basis GMRES implementation*, *IMA J. Numer. Anal.*, 14 (1994), pp. 563–581.
- [5] D. CALVETTI, J. PETERSEN AND L. REICHEL, *A parallel implementation of the GMRES algorithm*, in *Numerical Linear Algebra*, eds. L. Reichel, A. Ruttan and R. S. Varga, de Gruyter, Berlin, 1993, pp. 31–46.
- [6] D. CALVETTI, L. REICHEL AND D. C. SORENSEN, *An implicitly restarted Lanczos method for large symmetric eigenvalue problems*, *Elec. Trans. Numer. Anal.*, 2 (1994), pp. 1–21.
- [7] A. CHAPMAN AND Y. SAAD, *Deflated and augmented Krylov subspace techniques*, Report UMSI 95/181, University of Minnesota, Supercomputer Institute, Minneapolis, MN, 1995.
- [8] R. D. DA CUNHA AND T. HOPKINS, *A parallel implementation of the restarted GMRES iterative algorithm for nonsymmetric systems of linear equations*, *Adv. Comput. Math.*, 2 (1994), pp. 261–277.
- [9] J. DRKOSOVA, A. GREENBAUM, M. ROZLOZNIK, AND Z. STRAKOS, *Numerical stability of GMRES*, *BIT*, 35 (1995), pp. 309–320.
- [10] J. ERHEL, *A parallel GMRES version for general sparse matrices*, *Elec. Trans. Numer. Anal.*, 3 (1995), pp. 160–176.
- [11] J. ERHEL, K. BURRAGE AND B. POHL, *Restarted GMRES preconditioned by deflation*, *J. Comput. Appl. Math.*, to appear.
- [12] R. W. FREUND, G. H. GOLUB AND N. M. NACHTIGAL, *Iterative solution of linear systems*, *Acta Mathematica* 1992, (1991), pp. 57–100.
- [13] G. H. GOLUB AND C. F. VAN LOAN, *Matrix Computations*, 2nd ed., Johns Hopkins University Press, Baltimore, 1989.
- [14] A. GREENBAUM AND L. N. TREFETHEN, *GMRES/CR and Arnoldi/Lanczos as matrix approximation problems*, *SIAM J. Sci. Comput.*, 15 (1994), pp. 359–368.
- [15] M. J. GROTE AND T. HUCKLE, *Parallel preconditioning with sparse approximate inverses*, *SIAM J. Sci. Comput.*, 18 (1997), to appear.
- [16] S. A. KHARCHENKO AND A. YU. YEREMIN, *Eigenvalue translation based preconditioners for the GMRES(k) method*, *Numer. Linear Algebra Appl.*, 2 (1995), pp. 51–77.
- [17] R. LEHOUCQ, *Analysis and implementation of an implicitly restarted Arnoldi iteration*, Ph.D. Thesis, Rice University, Houston, 1995.
- [18] R. B. MORGAN, *A restarted GMRES method augmented with eigenvectors*, *SIAM J. Matrix Anal. Appl.*, 16 (1995), pp. 1154–1171.
- [19] N. M. NACHTIGAL, L. REICHEL AND L. N. TREFETHEN, *A hybrid GMRES algorithm for nonsymmetric linear systems*, *SIAM J. Matrix Anal. Appl.*, 13 (1992), pp. 796–825.
- [20] Y. SAAD, *Krylov subspace methods for solving large unsymmetric systems of linear equations*, *Math. Comp.*, 37 (1981), pp. 105–126.
- [21] Y. SAAD, *Preconditioned Krylov subspace methods for CFD applications*, Report UMSI 94/171, University of Minnesota, Supercomputer Institute, Minneapolis, MN, 1994.
- [22] Y. SAAD AND M. H. SCHULTZ, *GMRES: a generalized minimum residual algorithm for solving non-symmetric linear systems*, *SIAM J. Sci. Stat. Comput.*, 7 (1986), pp. 856–869.
- [23] D. C. SORENSEN, *Implicit application of polynomial filters in a k-step Arnoldi method*, *SIAM J. Matrix Anal. Appl.*, 13 (1992), pp. 357–385.
- [24] J. STOER AND R. BULIRSCH, *Introduction to Numerical Analysis*, Springer, New York, 1980.
- [25] H. A. VAN DER VORST AND C. VUIK, *The superlinear convergence behaviour of GMRES*, *J. Comput. Appl. Math.*, 48 (1993), pp. 327–341.
- [26] H. F. WALKER AND L. ZHOU, *A simpler GMRES*, *Numer. Linear Algebra Appl.*, 1 (1994), pp. 571–581.

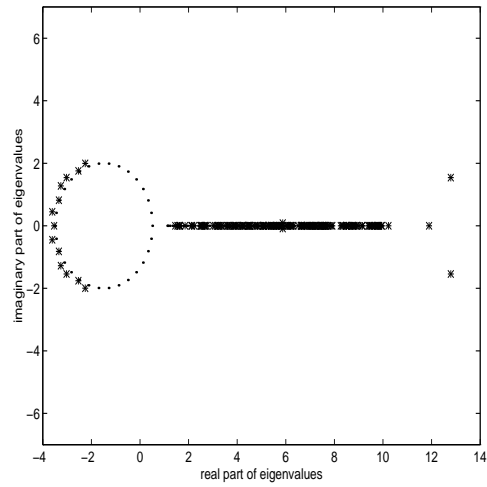


FIG. 4.1. Eigenvalues of A (\cdot) and $M^{-1}A$ ($*$) for Example 4.1 using Algorithm 3.5 with shifts (2.10).

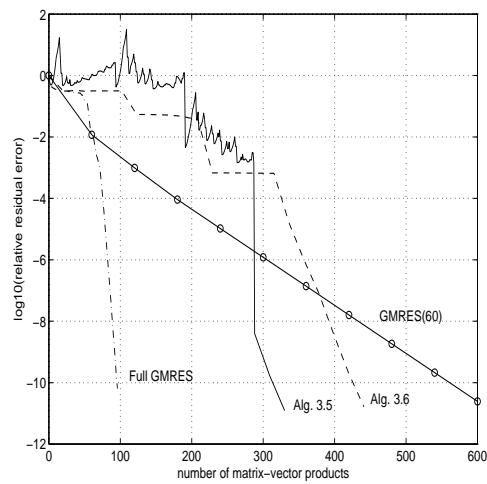


FIG. 4.2. Convergence for Example 4.1.

TABLE 4.1
Example 4.1.

method	size of Krylov subspace	# matrix-vector products	# preconditioners	# vectors in each preconditioner	total # vectors used
Algorithm 3.5	20	331	3	10	54
Algorithm 3.6	20	442	3	10	54
GMRES(60)	60	600	0	0	62
Full GMRES	96	96	0	0	96

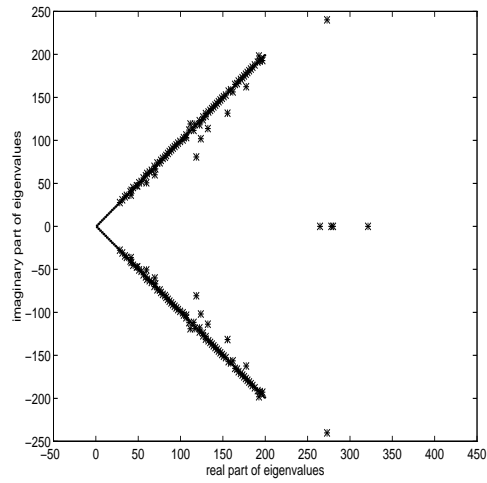


FIG. 4.3. Eigenvalues of A (\cdot) and $M^{-1}A$ ($*$) for Example 4.2 using Algorithm 3.5 with shifts (2.10).

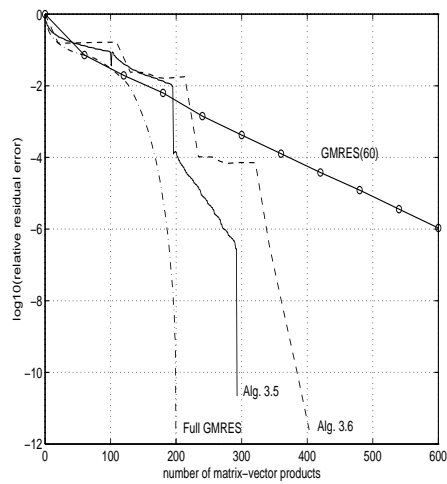


FIG. 4.4. Convergence for Example 4.2.

TABLE 4.2
Example 4.2.

method	size of Krylov subspace	# matrix-vector products	# preconditioners	# vectors in each preconditioner	total # vectors used
Algorithm 3.5	20	294	3	10	54
Algorithm 3.6	20	406	3	10	54
GMRES(60)	60	1080	0	0	62
Full GMRES	200	200	0	0	203

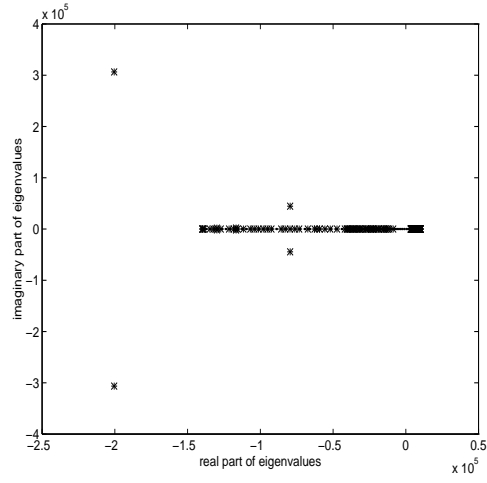


FIG. 4.5. Eigenvalues of A (\cdot) and $M^{-1}A$ ($*$) for Example 4.3 using Algorithm 3.5 with shifts (2.10).

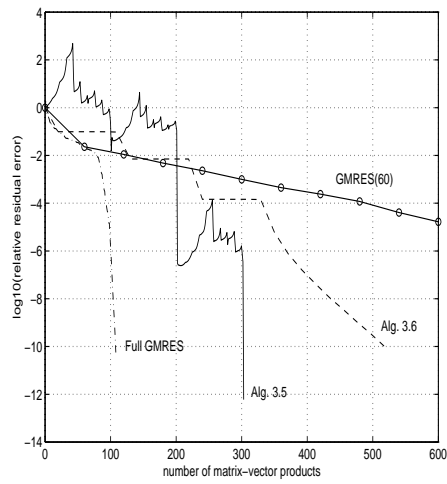


FIG. 4.6. Convergence for Example 4.3.

TABLE 4.3
Example 4.3.

method	size of Krylov subspace	# matrix-vector products	# preconditioners	# vectors in each preconditioner	total # vectors used
Algorithm 3.5	20	304	3	10	54
Algorithm 3.6	20	519	3	10	54
GMRES(60)	60	1560	0	0	62
Full GMRES	108	108	0	0	111

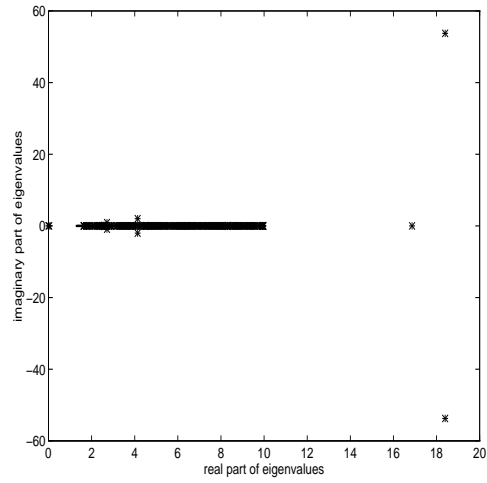


FIG. 4.7. Eigenvalues of A (\cdot) and $M^{-1}A$ ($*$) for Example 4.4 using Algorithm 3.5 with shifts (2.10).

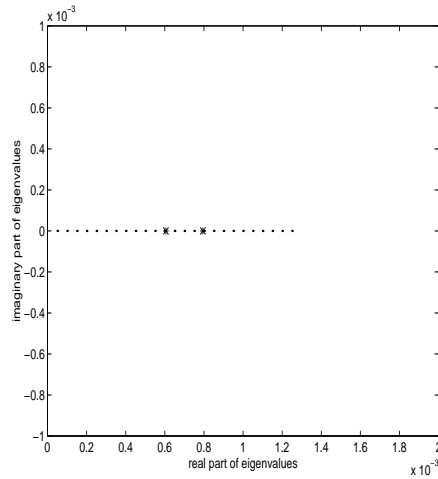


FIG. 4.8. The smallest eigenvalues of A (\cdot) and $M^{-1}A$ ($*$) for Example 4.4 using Algorithm 3.5 with shifts (2.10).

TABLE 4.4
Example 4.4.

method	size of Krylov subspace	# matrix-vector products	# preconditioners	# vectors in each preconditioner	total # vectors used
Algorithm 3.5	20	430	3	10	54
Algorithm 3.6	20	414	3	10	54
GMRES(60)	60	960	0	0	62
Full GMRES	200	200	0	0	203

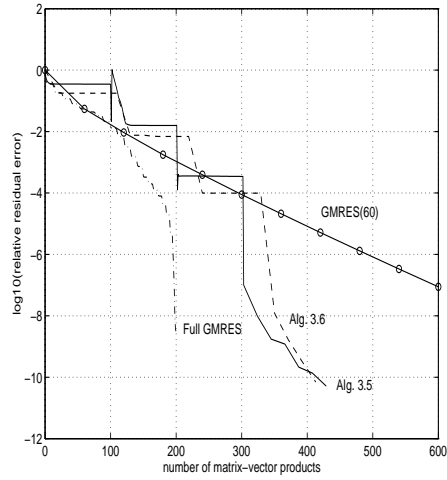


FIG. 4.9. *Convergence for Example 4.4.*

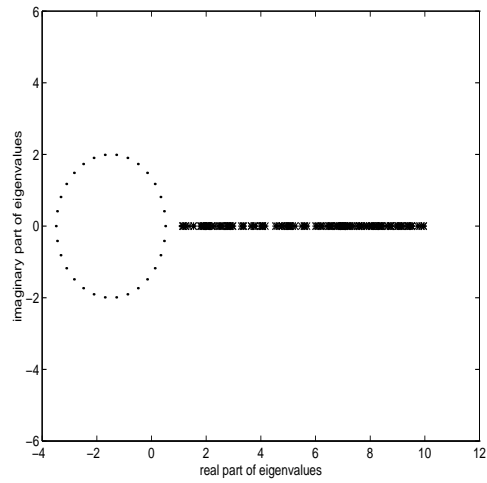


FIG. 4.10. *Eigenvalues of A (\cdot) and $M^{-1}A$ ($*$) for Example 4.5 using Algorithm 3.6 with eigenvalues of H_m with largest real part chosen as shifts.*