SIAM J. MATRIX ANAL. APPL.
Vol. 30, No. 1, pp. 102–120

THE ARNOLDI PROCESS AND GMRES FOR NEARLY SYMMETRIC MATRICES*

BERNHARD BECKERMANN† AND LOTHAR REICHEL‡

Abstract. Matrices with a skew-symmetric part of low rank arise in many applications, including path following methods and integral equations. This paper explores the properties of the Arnoldi process when applied to such a matrix. We show that an orthogonal Krylov subspace basis can be generated with short recursion formulas and that the Hessenberg matrix generated by the Arnoldi process has a structure, which makes it possible to derive a progressive GMRES method. Eigenvalue computation is also considered.

Key words. low-rank perturbation, iterative method, solution of linear system, eigenvalue computation

AMS subject classifications. 65F10, 65F15

DOI. 10.1137/060668274

1. Introduction.

This paper discusses the Arnoldi process applied to a large matrix $A \in \mathbb{R}^{n\times n}$ with a skew-symmetric part

$$A - A^* = \sum_{k=1}^{s} f_k g_k^*, \qquad f_k, g_k \in \mathbb{R}^n, \tag{1.1}$$

of low rank s. In particular, we assume that $s \ll n$. The superscript $*$ denotes transposition and, when applicable, complex conjugation. We present our results for matrices A and vectors f_k and g_k with real entries; however, our algorithms also can be applied to matrices and vectors with complex entries.

Linear systems of equations

$$Ax = b \tag{1.2}$$

with large matrices of this kind arise in path following methods, from integral equations as well as from certain boundary value problems for partial differential equations.

The generalized minimal residual (GMRES) method is one of the most popular iterative methods for the solution of large linear systems of equations with a nonsymmetric matrix. The standard implementation of GMRES is based on the Arnoldi process; see, e.g., Saad [15, section 6.5]. Application of j steps of the Arnoldi process to the matrix A with initial vector $r_0 \neq 0$ yields the decomposition

$$AV_j = V_j H_j + h_j e_j^*, \tag{1.3}$$

where $V_j = [v_1, v_2, \ldots, v_j] \in \mathbb{R}^{n\times j}$ and $h_j \in \mathbb{R}^n$ satisfy $V_j^* V_j = I_j$, $V_j^* h_j = 0$, and $v_1 = r_0/\|r_0\|$. Moreover, $H_j \in \mathbb{R}^{j\times j}$ is an upper Hessenberg matrix. Throughout

*Received by the editors August 24, 2006; accepted for publication (in revised form) by M. Benzi July 2, 2007; published electronically February 6, 2008.
http://www.siam.org/journals/simax/30-1/66827.html

†Laboratoire Painlevé UMR 8524 (ANO-EDP), UFR Mathématiques – M3, UST Lille, F-59655 Villeneuve d'Ascq CEDEX, France (bbecker@math.univ-lille1.fr). This author's research was supported in part by INTAS research network NeCCA 03-51-6637.

‡Department of Mathematical Sciences, Kent State University, Kent, OH 44242 (reichel@math.kent.edu). This author's research was supported in part by NSF grant DMS-0107858 and an OBR Research Challenge Grant.

this paper I_j denotes the identity matrix of order j, e_k denotes the kth column of an identity matrix of appropriate order, and $\|\cdot\|$ denotes the Euclidean vector norm. For ease of discussion, we will assume that j is small enough so that the decomposition (1.3) with the stated properties exists.

When $h_j \neq 0$, we can express (1.3) in the form

$$AV_j = V_{j+1}\bar{H}_j, \tag{1.4}$$

where $v_{j+1} = h_j/\|h_j\|$ and

$$V_{j+1} = [V_j, v_{j+1}] \in \mathbb{R}^{n\times(j+1)}, \qquad \bar{H}_j = \begin{bmatrix} H_j \\ \|h_j\|e_j^* \end{bmatrix} \in \mathbb{R}^{(j+1)\times j}.$$

The computation of the Arnoldi decompositions (1.3) or (1.4) of a general $n \times n$ matrix A requires the evaluation of j matrix-vector products with A and of about $j^2/2$ inner products with n-vectors. The latter demands $\mathcal{O}(nj^2)$ arithmetic floating point operations (flops) and may dominate the computational work. The Arnoldi process determines the columns of V_j in order and requires access to all the previously generated columns to compute the next one; in particular, all the columns of V_j have to be stored; see, e.g., Saad [15, section 6.3] for a thorough treatment of the Arnoldi process. Computation of the jth iterate by GMRES also requires the whole matrix V_j to be available. To limit the demand of computer memory, GMRES is often restarted periodically, say, every m steps. This restarted GMRES method is denoted by GMRES(m). Restarting may reduce the rate of convergence of GMRES significantly.

In section 2, we show that the property (1.1) of A makes it possible to determine the columns v_k of V_j with a short recursion formula, the number of terms of which depends on s in (1.1) but can be bounded independently of k. The recursion formula allows the computation of all the columns of V_j in only $\mathcal{O}(nj)$ flops. Moreover, the computation of v_k for large k does not require access to all the previously computed columns of V_j. Section 3 discusses the structure of the Hessenberg matrix H_j in (1.3) when A satisfies (1.1) and presents a fast algorithm for determining the Arnoldi decomposition (1.4).

The short recursion formula for the columns of V_j and the structure of H_j make it possible to derive a progressive GMRES method for the solution of linear systems (1.2) with a matrix that satisfies (1.1). Such a method is described in section 4. The storage requirement of the method, as well as the computational effort per iteration, are bounded independently of the number of iterations j. This makes it possible to apply the method without periodic restarts. Computed examples are presented in section 5 and concluding remarks can be found in section 6.

Recently, Barth and Manteuffel [4] presented iterative methods of conjugate gradient type for linear systems of equations of the kind considered in the present paper. Specifically, they considered linear systems of equations with a generalized B-normal(ℓ, m) matrix. This type of matrix is characterized by the existence of polynomials p_ℓ and q_m of degrees ℓ and m, respectively, such that the matrix

$$A^\dagger q_m(A) - p_\ell(A)$$

is of low rank, where $A^\dagger = B^{-1}A^*B$ and B is a Hermitian positive definite matrix. The matrix $A^\dagger$ is the adjoint of A with respect to the B-inner product

$$\langle u, v\rangle_B = u^*Bv.$$

In the terminology of Barth and Manteuffel [4] matrices A that satisfy (1.1) are generalized I-normal$(1,0)$ matrices.

Barth and Manteuffel [4] derived their methods by generalizing the recurrence relations for orthogonal polynomials on the unit circle. The latter type of recurrence relations had previously been applied to iterative methods in [11, 12]; see also Arnold et al. [2] for a recent application to QCD computations. The derivation of our iterative methods for (1.2) differs from the derivation by Barth and Manteuffel [4] of their schemes in that we do not apply properties of orthogonal polynomials on the unit circle. Iterative methods for linear systems of equations with a matrix, whose symmetric part is positive definite and easily invertible, are described by Concus and Golub [7] and Widlund [18].

2. Generation of an orthogonal Krylov subspace basis. Introduce the Krylov subspace

$$\mathcal{K}_j(A,b) = \text{span}\{b, Ab, A^2b, \ldots, A^{j-1}b\}, \tag{2.1}$$

which we assume to be of dimension j. The columns of the matrix V_j in (1.3) form an orthonormal basis of $\mathcal{K}_j(A,b)$.

Let f_k and g_k be the vectors in (1.1) and define the matrices

$$F = [f_1, f_2, \ldots, f_s], \qquad G = [g_1, g_2, \ldots, g_s], \tag{2.2}$$

which we may assume to be of full rank; otherwise we can reduce s. We express (1.1) as

$$A - A^* = FG^* \tag{2.3}$$

and note that

$$FG^* = -GF^*. \tag{2.4}$$

It follows from (2.4) and the fact that F and G are of full rank that s is even and that there is a unique matrix $C \in \mathbb{R}^{s\times s}$, such that

$$G = FC. \tag{2.5}$$

The fact that s is even can be seen by substituting (2.5) into (2.4). This yields $C^* = -C$. Therefore, when s is odd, C is singular and G is not of full rank. Use of the representation (2.5) of G reduces the computational work in the algorithms presented in sections 3 and 4.

Example 2.1. In many applications that involve a matrix A with a skew-symmetric part of low rank, the matrix is given in the form

$$A = M + \sum_{k=1}^{s/2} f_k g_k^*$$

with $M \in \mathbb{R}^{n\times n}$ symmetric. Then (1.1) can be expressed as

$$A - A^* = \sum_{k=1}^{s/2} f_k g_k^* - \sum_{k=1}^{s/2} g_k f_k^*$$

and we may choose

$$F = [f_1, f_2, \ldots, f_{s/2}, g_1, g_2, \ldots, g_{s/2}], \qquad C = \begin{bmatrix} 0 & -I_{s/2} \\ I_{s/2} & 0 \end{bmatrix}.$$

Introduce the vectors

$$f_{\ell,k} = V_k V_k^* f_\ell, \qquad 1 \le \ell \le s, \qquad 1 \le k \le j. \tag{2.6}$$

Then

$$f_{\ell,k} \in \mathcal{K}_k(A, r_0), \qquad f_{\ell,k} - f_\ell \perp \mathcal{K}_k(A, r_0). \tag{2.7}$$

Moreover, for each ℓ, the $f_{\ell,k}$ satisfy the recursion

$$\begin{cases} f_{\ell,k} = f_{\ell,k-1} + v_k v_k^* f_\ell, & k = 2, 3, \ldots, j, \\ f_{\ell,1} = v_1 v_1^* f_\ell. \end{cases} \tag{2.8}$$

Let

$$v_k' = A^* v_k + \sum_{\ell=1}^{s} g_\ell^* v_k (f_\ell - f_{\ell,k}). \tag{2.9}$$

Then (1.1) gives

$$v_k' - A v_k = \sum_{\ell=1}^{s} g_\ell^* v_k (f_\ell - f_{\ell,k}) - (A - A^*) v_k = -\sum_{\ell=1}^{s} g_\ell^* v_k f_{\ell,k}. \tag{2.10}$$

We may assume that $Av_k \notin \mathcal{K}_k(A, r_0)$, because otherwise range(V_k) is an invariant subspace of A, which contains the solution of the linear system (1.2); see, e.g., Saad [15, section 6.5.4] for details. The following properties of v_k' are a consequence of the above discussion.

PROPOSITION 2.2. *Let v_k' be defined by* (2.9) *and assume that* $\dim \mathcal{K}_{k+1}(A, r_0) = k + 1$. *Then*

$$v_k' \in \mathcal{K}_{k+1}(A, r_0) \setminus \mathcal{K}_k(A, r_0), \qquad v_k' \perp \mathcal{K}_{k-2}(A, r_0). \tag{2.11}$$

Proof. The requirement that $\mathcal{K}_{k+1}(A, r_0)$ be of dimension $k + 1$ secures that $Av_k \notin \mathcal{K}_k(A, r_0)$. Equation (2.7) yields that $v_k' - Av_k \in \mathcal{K}_k(A, r_0)$, and this establishes the left-hand side of (2.11).

It follows from the Arnoldi decomposition (1.3) that $v_k \perp Av_\ell$ for $1 \le \ell \le k - 2$, or, equivalently, that $A^* v_k \perp v_\ell$ for $1 \le \ell \le k-2$. The latter property, in combination with (2.7) and (2.9), shows the orthogonality relation (2.11). ∎

Equation (2.10) yields the expression

$$v_k' = Av_k - \sum_{\ell=1}^{s} g_\ell^* v_k f_{\ell,k}, \tag{2.12}$$

which we use to evaluate v_k'. Orthogonalization against the vectors v_{k-1} and v_k, and normalization of the resulting vector, gives the Arnoldi vector v_{k+1}. In what follows we will write this operation more explicitly as

$$v_k' = t_{k+1,k} v_{k+1} + t_{k,k} v_k + t_{k-1,k} v_{k-1}, \qquad k \ge 1, \tag{2.13}$$

where

$$t_{k-1,k} = v_{k-1}^* v_k', \quad t_{k,k} = v_k^* v_k', \quad t_{k+1,k} = v_{k+1}^* v_k', \tag{2.14}$$

with $v_0 = 0$ and $t_{k+1,k} = \|v_k' - t_{k,k} v_k - t_{k,k-1} v_{k-1}\| > 0$. The computations for generating the orthogonal Krylov subspace basis, and for determining the matrix $\bar{H}_j$ in (1.4), are summarized in Algorithm 3.2 of the following section.

3. Structure of the Hessenberg matrices. This section discusses the structure of the matrices $H_j = [h_{k,\ell}]$ and $\bar{H}_j = [h_{k,\ell}]$ in the Arnoldi decompositions (1.3) and (1.4), respectively. It is convenient to introduce the following terminology. For an integer m, the m-diagonal of a matrix $B = [b_{k,\ell}]$ consists of all entries of the form $b_{k,k+m}$. The m-upper (m-lower) triangular part of B is the submatrix comprising all entries on and above (below) the m-diagonal. For instance, the upper Hessenberg matrices H_j and $\bar{H}_j$ have vanishing (-2)-lower triangular parts. Note that the (-2)-upper triangular part is not triangular.

PROPOSITION 3.1. *Let $\hat{F}_j = V_j^* F$ and $\hat{G}_j = V_j^* G$, where F and G are the matrices in* (2.3) *and $j \geq s$. Then the upper Hessenberg matrix H_j in the Arnoldi decomposition* (1.3) *satisfies*

$$H_j - H_j^* = \hat{F}_j \hat{G}_j^*, \qquad \hat{G}_j \hat{F}_j^* = -\hat{F}_j \hat{G}_j^*, \tag{3.1}$$

i.e., H_j has a skew-symmetric part of rank s. Moreover, H_j and $\hat{F}_j \hat{G}_j^*$ have the same 2*-upper triangular parts.*

Proof. It follows from (1.3) and (2.3) that

$$H_j = V_j^* A V_j = V_j^* (A^* + FG^*) V_j = H_j^* + \hat{F}_j \hat{G}_j^*,$$

which shows (3.1). Since the (-2)-lower triangular part of H_j vanishes, (3.1) yields the 2-upper triangular part of H_j. ∎

The proposition shows that H_j is an order-$(1, s+1)$ quasi-separable matrix; see, e.g., Eidelman, Gohberg, and Olshevsky [9] for a recent discussion on this kind of matrix.

We turn to the entries in the tridiagonal part of $\bar{H}_j$. In accordance with (2.14), we define the matrix $\bar{T}_j = [t_{m,k}] \in \mathbb{R}^{(j+1)\times j}$ with entries $t_{m,k} = v_m^* v_k'$. Notice that $\bar{T}_j$ is tridiagonal by Proposition 2.2. Substitution of (2.6) into (2.12) gives

$$v_k' = Av_k - \sum_{\ell=1}^{s} (g_\ell^* v_k) V_k V_k^* f_\ell = Av_k - V_k V_k^* F G^* v_k = Av_k - V_k \hat{F}_k \hat{G}_k^* e_k,$$

and, taking into account that $e_m^* \hat{F}_k \hat{G}_k^* e_k = e_m^* \hat{F}_j \hat{G}_j^* e_k$ for $m \leq k \leq j$, we get for the entries $h_{m,k} = v_m^* A v_k$ of $\bar{H}_j$ the formula

$$h_{m,k} = \begin{cases} t_{k+1,k}, & m = k+1, \\ t_{m,k} + e_m^* \hat{F}_j \hat{G}_j^* e_k, & k-1 \leq m \leq k, \\ e_m^* \hat{F}_j \hat{G}_j^* e_k, & 1 \leq m < k-1. \end{cases} \tag{3.2}$$

Thus, the matrix $\hat{F}_j \hat{G}_j^*$ contributes to the upper triangular part of $\bar{H}_j$, and the matrix $\bar{T}_j$, which expresses the orthogonalization of the vectors v_k', contributes to the tridiagonal part; in MATLAB notation, we have

$$\bar{H}_j = \bar{T}_j + \mathsf{triu}(\hat{F}_j \hat{G}_j^*, 0).$$

Combining (2.9) with (2.7) yields

$$v_m^* v_k' = v_m^* A^* v_k = (v_k^* A v_m)^*, \qquad 1 \leq m \leq k,$$

and comparison with (2.14) gives

$$t_{k-1,k} = t_{k,k-1} > 0, \qquad t_{k,k} = h_{k,k}^*. \tag{3.3}$$

We describe an algorithm for the computation of the matrices $\bar{H}_j$ and V_{j+1} in the decomposition (1.4), assuming that the decomposition exists. The matrix $\bar{H}_j$ is represented in decomposed form (3.2) by the matrices $\hat{F}_j$, $\hat{G}_j$, and $\bar{T}_j$, which in the algorithm are represented without subscript j. The subscripts used in the algorithm denote row and column indices. Thus, $\hat{F}_{k,:}$ denotes the kth row of the matrix $\hat{F}_j$. At iteration k, we let $\tilde{F} = [f_{1,k}, f_{2,k}, \ldots, f_{s,k}]$.

ALGORITHM 3.2. *Generation of the matrices $\bar{H}_j$ and V_{j+1}.*

Input: $A \in \mathbb{R}^{n\times n}$, $F = [f_1, f_2, \ldots, f_s], G = [g_1, g_2, \ldots, g_s] \in \mathbb{R}^{n\times s}$, $r_0 \in \mathbb{R}^n$, j;

Output: $\bar{T} = [t_{\ell,k}] \in \mathbb{R}^{(j+1)\times j}$, $\hat{F}, \hat{G} \in \mathbb{R}^{(j+1)\times s}$, $V_{j+1} = [v_1, v_2, \ldots, v_{j+1}] \in \mathbb{R}^{n\times(j+1)}$;

1. $\tilde{F} := 0$;
2. $v_1 := r_0/\|r_0\|$;
3. *for* $k = 1 : j$
   4. $\hat{F}_{k,:} := v_k^* F$; $\hat{G}_{k,:} := v_k^* G$;
   5. $\tilde{F} := \tilde{F} + v_k \hat{F}_{k,:}$;
   6. $v' := Av_k - \tilde{F}\hat{G}_{k,:}^*$;
   7. *if* $k > 1$ *then*
      8. $t_{k-1,k} := v_{k-1}^* v'$; $v' := v' - t_{k-1,k} v_{k-1}$;
   9. *endif*
   10. $t_{k,k} := v_k^* v'$; $v' := v' - t_{k,k} v_k$; $t_{k+1,k} := \|v'\|$; $v_{k+1} := v'/t_{k+1,k}$;
11. *endfor*

We note that the computational effort of line 4 of the algorithm can be essentially halved by using the representation (2.5) of G.

Algorithm 3.2 can be applied to compute approximations of a few extreme eigenvalues and associated eigenvectors of A similarly to the standard implementation of the Arnoldi process. Certain eigenvalues of H_j are used to approximate selected eigenvalues of A. The structure of H_j therefore is of interest.

Remark 3.3. Given a unitary matrix $Q \in \mathbb{C}^{j\times j}$, it follows from Proposition 3.1 that for the matrix

$$S = Q^* H_j Q,$$

we have $\Sigma := S - S^* = Q^* \hat{F}_j \hat{G}_j^* Q$, i.e., S has a skew-symmetric part of rank s. If S has an additional sparsity structure, then we may derive results similarly to Proposition 3.1. For instance, the matrix S in the Schur normal form of H_j is upper triangular, and thus S may be written as a diagonal matrix plus the 1-upper triangular part of the matrix Σ. Similarly, the matrix S obtained after one step of the QR-algorithm is upper Hessenberg and therefore may be written as a tridiagonal matrix plus the 2-upper triangular part of the matrix Σ.

We recall that in the QR-algorithm for eigenvalue computations the unitary factor Q is chosen such that $R = Q^* H_j$ is upper triangular.

Remark 3.4. Consider the QR-decomposition $H_j = QR$ with orthogonal Q and upper triangular R. Here also the matrix R has a structure: since Q^* is known to be of lower Hessenberg form (see, e.g., the considerations of the next section), we see from Proposition 3.1 that the 3-upper triangular part of $Q^*(H_j - \hat{F}_j \hat{G}_j)$ contains only zeros, or, in other words, the 3-upper triangular parts of R_j and of the matrix $Q^* \hat{F}_j \hat{G}_j$ of rank s coincide.

The structure makes it possible to compute the matrix R in $\mathcal{O}(j)$ flops, by representing H_j in terms of the tridiagonal part of H_j and the matrices $\hat{F}_j$ and $\hat{G}_j$, and by representing R in terms of its 0-, 1-, and 2-diagonals and the matrices $Q^* \hat{F}_j$ and $\hat{G}_j$.

Since the computation of R does not play a role in subsequent considerations, we omit the details.

4. A progressive GMRES algorithm. Let $x_0 \in \mathbb{R}^n$ be an approximate solution of (1.2). GMRES determines a new approximate solution x_j of (1.2), such that

$$\|Ax_j - b\| = \min_{x \in x_0 + \mathcal{K}_j(A,r_0)} \|Ax - b\|, \qquad x_j \in x_0 + \mathcal{K}_j(A, r_0). \tag{4.1}$$

The standard implementation of GMRES determines a correction of x_0, i.e., $x_j = x_0 + V_j x_j$, by substituting the decomposition (1.4) with $r_0 = b - Ax_0$ into (4.1); see, e.g., Saad [15, section 6.5] for details. This gives the equivalent minimization problem

$$\min_{y \in \mathbb{R}^j} \|\bar{H}_j y - e_1 \|r_0\| \|, \tag{4.2}$$

with solution $y_j \in \mathbb{R}^j$.

We solve the least-squares problem (4.2) by using the QR-factorization $\bar{H}_j = Q_{j+1}\bar{R}_j$, where $Q_{j+1} \in \mathbb{R}^{(j+1)\times(j+1)}$ is orthogonal (or unitary in the case of complex A, b) and

$$\bar{R}_j = \begin{bmatrix} R_j \\ 0 \end{bmatrix} \in \mathbb{R}^{(j+1)\times j}, \tag{4.3}$$

with $R_j \in \mathbb{R}^{j\times j}$ upper triangular. Let us first recall in the following paragraph and Proposition 4.1 the well-known construction of a QR-decomposition of the upper Hessenberg matrix $\bar{H}_j$ for a general matrix A. Subsequently, we explain in Proposition 4.2 how the structure of the matrix A helps us to derive a progressive form of GMRES.

Following Saad [15, Chapter 6.5.3], we determine the matrix Q_{j+1} by applying a product of Givens rotations to $\bar{H}_j$. Let $Q_1 = [1]$ and define, for $k = 1, 2, \ldots, j$,

$$Q_{k+1}^* = \Omega_{k+1} \begin{bmatrix} Q_k^* & 0 \\ 0 & 1 \end{bmatrix}, \qquad \Omega_{k+1} = \begin{bmatrix} I_{k-1} & 0 & 0 \\ 0 & c_k^* & s_k \\ 0 & -s_k & c_k \end{bmatrix}, \tag{4.4}$$

with $s_k \geq 0$ and $s_k^2 + |c_k|^2 = 1$ such that Ω_{k+1} is unitary (and reduces to a classical Givens rotation in the case of real data). Using the nested structure of $\bar{H}_j = [h_{k,\ell}]$, i.e., the fact that $\bar{H}_{j-1}$ is the leading $j \times (j-1)$ principal submatrix of $\bar{H}_j$, yields

$$Q_{j+1}^* \bar{H}_j = \Omega_{j+1} \begin{bmatrix} Q_j^* \bar{H}_{j-1} & Q_j^* H_j e_j \\ 0 & h_{j+1,j} \end{bmatrix} = \Omega_{j+1} \begin{bmatrix} R_{j-1} & * \\ 0 & \tau_j \\ 0 & h_{j+1,j} \end{bmatrix},$$

with

$$\tau_j = e_j^* Q_j^* H_j e_j. \tag{4.5}$$

Since multiplication by Ω_{j+1} affects only the last two rows, the matrices R_j and $\bar{R}_j$ also have a nested substructure:

$$\bar{R}_j = \begin{bmatrix} \bar{R}_{j-1} & * \\ 0 & 0 \end{bmatrix}, \qquad R_j = \begin{bmatrix} R_{j-1} & * \\ 0 & * \end{bmatrix}.$$

We have the following formulas for the coefficients c_j, s_j of Ω_{j+1} and for the entries of Q^*_{j+1}.

PROPOSITION 4.1. *There holds*

$$s_j = \frac{t_{j+1,j}}{\sqrt{t^2_{j+1,j} + |\tau_j|^2}} \geq 0, \qquad c_j = \frac{\tau_j}{\sqrt{t^2_{j+1,j} + |\tau_j|^2}}, \tag{4.6}$$

where $t_{j+1,j} = h_{j+1,j}$ is the last subdiagonal entry of $\bar{H}_j$ and τ_j is given by (4.5). *The first j rows of Q^*_{j+1} are obtained by padding a zero on the right-hand side of the corresponding rows of Q^*_j. In particular, Q^*_{j+1} is of lower Hessenberg form, with its lower triangular part coinciding with the lower triangular part of a rank-one matrix. Moreover, for $j \geq 3$,*

$$e^*_{j+1} Q^*_{j+1} = [\, -s_j e^*_j Q^*_j,\ c_j \,] = [\, *\ ,\ s_j s_{j-1} c_{j-2}\ ,\ -s_j c_{j-1}\ ,\ c_j \,]. \tag{4.7}$$

Proof. The proof is obtained by direct calculations. ∎

We are in a position to describe a progressive recurrence relation for the GMRES residual r_j, a simplified recurrence for its norm, as well as a simplified expression for the quantity τ_j defined by (4.5). In particular, the progressive GMRES algorithm does not require the entries of the matrices R_j, $\bar{H}_j$, and Q_{j+1}. Only the c_k, s_k of the Givens rotations (4.4) and the quantities occurring in the recurrence relation for the Arnoldi vectors v_k are needed.

PROPOSITION 4.2. *Let r_j denote the residual vector associated with x_j, i.e.,*

$$r_j = b - Ax_j, \tag{4.8}$$

and define recursively

$$\gamma_j = -s_j \gamma_{j-1}, \qquad j \geq 1, \tag{4.9}$$

where $\gamma_0 = \|r_0\|$. Then $\gamma_j = (-1)^j \|r_j\|$. Moreover,

$$r_j = s^2_j r_{j-1} + \gamma_j c^*_j v_{j+1}, \qquad j \geq 1. \tag{4.10}$$

Finally, define the vectors $p_j \in \mathbb{R}^s$ recursively by

$$p^*_j = -s_{j-1} p^*_{j-1} + c_{j-1} e^*_j \hat{F}_j, \qquad j \geq 2, \tag{4.11}$$

and $p^*_1 = \hat{F}_1$. Then we get for the scalar τ_j defined by (4.5) *the expression*

$$\tau_j = c_{j-1} t_{j,j} - s_{j-1} c_{j-2} t_{j-1,j} + p^*_j \hat{G}^*_j e_j, \qquad j \geq 2, \tag{4.12}$$

with $c_0 = 1$ and $\tau_1 = t^*_{1,1}$.

Proof. We start by establishing the formula

$$r_j = \gamma_j V_{j+1} Q_{j+1} e_{j+1}. \tag{4.13}$$

A different proof is presented by Saad [15, Proposition 6.9]. From the definition of GMRES, we have that $r_j = P^{\perp}_{A\mathcal{K}_j(A,r_0)} r_0$, where $P_{A\mathcal{K}_j(A,r_0)}$ denotes the orthogonal projector onto $A\mathcal{K}_j(A, r_0)$ and $P^{\perp}_{A\mathcal{K}_j(A,r_0)} = I - P_{A\mathcal{K}_j(A,r_0)}$ denotes the orthogonal projector onto the complement. Denote by $\bar{Q}_j \in \mathbb{R}^{(j+1)\times j}$ the matrix made up of the first j columns of Q_{j+1}. From (1.4) and (4.3), we obtain that

$AV_j = V_{j+1}Q_{j+1}\bar{R}_j = V_{j+1}\bar{Q}_j R_j$. Since R_j is invertible, we see that an orthonormal basis of $A\mathcal{K}_j(A, r_0)$ is given by the columns of $V_{j+1}\bar{Q}_j$, implying that

$$\begin{aligned} r_j &= r_0 - P_{A\mathcal{K}_j(A,r_0)}r_0 = V_{j+1}Q_{j+1}Q_{j+1}^*V_{j+1}^*r_0 - V_{j+1}\bar{Q}_j\bar{Q}_j^*V_{j+1}^*r_0 \\ &= V_{j+1}(Q_{j+1}Q_{j+1}^* - \bar{Q}_j\bar{Q}_j^*)e_1\|r_0\| = V_{j+1}Q_{j+1}e_{j+1}e_{j+1}^*Q_{j+1}^*e_1\|r_0\|. \end{aligned}$$

It follows from (4.7) and (4.9) that

$$\gamma_0\, e_{j+1}^*Q_{j+1}^*e_1 = \gamma_0(-s_j)\, e_j^*Q_j^*e_1 = \cdots = \gamma_0(-s_j)(-s_{j-1})\dots(-s_1) = \gamma_j.$$

This establishes (4.13). Since $V_{j+1}Q_{j+1}$ has orthonormal columns and $s_k \geq 0$ by Proposition 4.1, we may conclude by taking norms in (4.13) that $|\gamma_j| = \|r_j\| = (-1)^j\gamma_j$.

The updating formula (4.10) is now an immediate consequence of (4.13): by (4.7),

$$r_j = \gamma_j V_{j+1}[-s_j e_j^*Q_j^*, c_j]^* = -s_j\frac{\gamma_j}{\gamma_{j-1}}r_{j-1} + \gamma_j c_j^* v_{j+1}.$$

It remains to show (4.12). From (4.7) and (4.11) we conclude by recurrence on j that

$$p_j^* = e_j^*Q_j^*\hat{F}_j, \qquad j \geq 1.$$

The structure of H_j, together with (4.7) and (4.13), yields for $j \geq 2$ that

$$\begin{aligned} \tau_j &= e_j^*Q_j^*H_je_j \\ &= e_j^*Q_j^*\left(\begin{bmatrix} 0 \\ \vdots \\ 0 \\ t_{j-1,j} \\ t_{j,j} \end{bmatrix} + \hat{F}_j\hat{G}_j^*e_j\right) = [-s_{j-1}c_{j-2}, c_{j-1}]\begin{bmatrix} t_{j-1,j} \\ t_{j,j} \end{bmatrix} + p_j^*\hat{G}_j^*e_j. \end{aligned}$$

When $j = 1$, we get by using $Q_1 = [1]$ and (3.3) that $\tau_1 = h_{1,1} = t_{1,1}^*$. ∎

By applying a suitable linear operator L, such that $Lr_k = x_k$ for $0 \leq k \leq j+1$, to the recurrence relation (4.10) of the residuals, we obtain an updating formula for the GMRES iterates in terms of the auxiliary vectors $z_k = Lv_k$ and $w_{\ell,k} = Lf_{\ell,k}$, which together with the recursive computation of these new vectors is described in the following proposition.

PROPOSITION 4.3. *Let* $\dim\mathcal{K}_{j+1}(A, r_0) = j+1$ *and define recursively*

$$w_{\ell,k} = w_{\ell,k-1} + v_k^*f_\ell z_k, \quad 0 < k \leq j, \tag{4.14}$$

$$z_{k+1} = -\frac{1}{t_{k+1,k}}\left(v_k + t_{k,k}z_k + t_{k-1,k}z_{k-1} + \sum_{\ell=1}^{s} g_\ell^*v_k w_{\ell,k}\right), \quad 1 < k \leq j, \tag{4.15}$$

together with the initializations

$$w_{\ell,0} = 0, \qquad z_1 = \frac{x_0}{\gamma_0}, \qquad z_2 = -\frac{1}{t_{2,1}}(v_1 + t_{1,1}^*z_1). \tag{4.16}$$

Then we have for $0 < k \leq j$ *the updating formula*

$$x_k = s_k^2x_{k-1} + \gamma_k c_k^*z_{k+1}. \tag{4.17}$$

Proof. Consider the QR-factorization

$$[r_0, Ar_0, \ldots, A^j r_0] = V_{j+1} S_{j+1},$$

i.e., $S_{j+1} \in \mathbb{R}^{(j+1)\times(j+1)}$ is upper triangular and invertible by assumption on j. The projector

$$P = V_{j+1} S_{j+1} (I_{j+1} - e_1 e_1^*) S_{j+1}^{-1} V_{j+1}^*$$

satisfies

$$P\left(\sum_{k=0}^{j} \alpha_k A^k r_0\right) = \sum_{k=1}^{j} \alpha_k A^k r_0, \qquad (I-P)\left(\sum_{k=0}^{j} \alpha_k A^k r_0\right) = \alpha_0 r_0.$$

As a consequence, defining the linear operator L by

$$Lv = \frac{r_0^*(I-P)v}{r_0^* r_0} x_0 - A^{-1} P v,$$

we get for any $u \in \mathcal{K}_j(A, r_0)$ that

$$L(b - A(x_0 + u)) = x_0 + u.$$

In particular, we obtain $Lr_k = x_k$ for $0 \le k \le j$, as claimed above. In order to see that the vectors z_{k+1} and $w_{\ell,k}$ defined by

$$z_{k+1} = Lv_{k+1}, \qquad w_{\ell,k} = Lf_{\ell,k}, \qquad 0 \le k \le j,$$

can be computed via the relations (4.14)–(4.16), we argue by recurrence on k: applying L to the relations $f_{\ell,0} = 0$, $v_1 = r_0/\gamma_0 = (b - Ax_0)/\gamma_0$, and $Av_1 = h_{2,1}v_2 + h_{1,1}v_1 = t_{2,1}v_2 + t_{1,1}^* v_1$, respectively, leads to the initializations (4.16). Similarly, for (4.14) we apply L to (2.8), and (4.15) is obtained by applying L both to (2.12) and (2.13), where we notice that $L(Av_k) = -v_k$. Finally, the recurrence relation (4.17) for the GMRES iterates follows by applying L to (4.10). ∎

Let $W_j = [w_{1,j}, w_{2,j}, \ldots, w_{s,j}] \in \mathbb{R}^{n\times s}$. Then (4.14) can be written as

$$W_j = W_{j-1} + z_j e_j^* \hat{F}_j, \qquad W_1 = \frac{x_0}{\gamma_0}\hat{F}_1,$$

and

$$\sum_{\ell=1}^{s} g_\ell^* v_j \, w_{\ell,j} = W_j \hat{G}_j^* e_j.$$

Algorithm 4.4 below works with the matrices W_j rather than with their columns individually. The notation of Algorithm 4.4 follows that of Algorithm 3.2. In particular, the matrices W_j are stored in W.

ALGORITHM 4.4. *Progressive GMRES.*
Input: $A \in \mathbb{R}^{n\times n}$, $F = [f_1, f_2, \ldots, f_s], G = [g_1, g_2, \ldots, g_s] \in \mathbb{R}^{n\times s}$, $b, x_0 \in \mathbb{R}^n$;
Output: GMRES iterates $x_j \in \mathbb{R}^n$;
% initialization
1. $r_0 := b - Ax_0$; $\gamma_0 := \|r_0\|$;
2. $v_1 := r_0/\gamma_0$; $z_1 := x_0/\gamma_0$;

$\%\ j = 1$
3. $\hat{F}_{1,:} := v_1^* F$; $\hat{G}_{1,:} := v_1^* G$;
4. $p_1^* := \hat{F}_{1,:}$; $\tilde{F} := v_1 \hat{F}_{1,:}$; $W := x_0 \hat{F}_1/\gamma_0$;
5. $v' := Av_1 - \tilde{F}\hat{G}_{1,:}^*$;
6. $t_{1,1} := v_1^* v'$; $v' := v' - t_{1,1} v_1$;
7. $t_{2,1} := \|v'\|$; $v_2 := v'/t_{2,1}$;
8. $\tau_1 := t_{1,1}^*$;
9. $c_1 := \tau_1/(|\tau_1|^2 + t_{2,1}^2)^{1/2}$; $s_1 := t_{2,1}/(|\tau_1|^2 + t_{2,1}^2)^{1/2}$; $\gamma_1 := -s_1\gamma_0$;
10. $z_2 := -(v_1 + t_{1,1}^* z_1)/t_{2,1}$; $x_1 := s_1^2 x_0 + \gamma_1 c_1^* z_2$;
$\%\ j > 1$
11. *for* $j = 2, 3, \ldots$ *until convergence*
    12. $\hat{F}_{j,:} := v_j^* F$; $\hat{G}_{j,:} := v_j^* G$;
    13. $p_j^* := -s_{j-1} p_{j-1}^* + c_{j-1} \hat{F}_{j,:}$; $\tilde{F} := \tilde{F} + v_j \hat{F}_{j,:}$; $W := W + z_j \hat{F}_{j,:}$;
    14. $v' := Av_j - \tilde{F}\hat{G}_{j,:}^*$;
    15. $t_{j-1,j} := v_{j-1}^* v'$; $v' := v' - t_{j-1,j} v_{j-1}$;
    16. $t_{j,j} := v_j^* v'$; $v' := v' - t_{j,j} v_j$; $t_{j+1,j} := \|v'\|$; $v_{j+1} := v'/t_{j+1,j}$;
    17. $\tau_j := c_{j-1} t_{j,j} - s_{j-1} c_{j-2} t_{j-1,j} + p_j^* \hat{G}_{j,:}^*$;
    18. $c_j := \tau_j/(|\tau_j|^2 + t_{j+1,j}^2)^{1/2}$; $s_j := t_{j+1,j}/(|\tau_j|^2 + t_{j+1,j}^2)^{1/2}$; $\gamma_j := -s_j\gamma_{j-1}$;
    19. $z_{j+1} := -(v_j + t_{j,j} z_j + t_{j-1,j} z_{j-1} + W\hat{G}_{j,:}^*)/t_{j+1,j}$;
    20. $x_j := s_j^2 x_{j-1} + \gamma_j c_j^* z_{j+1}$;
21. *endfor*

Iterations with GMRES are typically terminated when the residual vector (4.8) is sufficiently small, e.g., when

$$\|r_j\|/\|r_0\| \leq \varepsilon \tag{4.18}$$

for a user-specified value of ε. This stopping criterion can be easily evaluated, since Algorithm 4.4 computes γ_j, with $|\gamma_j| = \|r_j\|$, in each iteration. If the residual vectors are desired in each iteration, then one can add the relation (4.10) on line 10 (for $j = 1$) and on line 20 of the algorithm. Stopping criteria of the type (4.18) have recently been discussed by Paige et al. [13, 14]. In particular, the initial vector x_0 should be chosen so that $\|r_0\| \leq \|b\|$ and preferably as the zero-vector.

In order to make the connection between Algorithm 4.4 and the preceding discussion clearer, vectors are equipped with subscripts in the algorithm. However, only the most recently generated vectors p_j^* and x_j have to be stored simultaneously, and only the two most recently generated vectors v_j, v_{j-1} and z_j, z_{j-1} have to be stored at any given time. Only the jth rows of the matrices $\hat{F}$ and $\hat{G}$ have to be stored simultaneously. The matrices $\tilde{F}$ and W have to be stored and require $n \times s$ storage locations each. Moreover, representations of the matrices A, F, and G have to be stored. Ignoring the storage for the latter, the storage requirement for Algorithm 4.4 is bounded by $(2s+6)n + \mathcal{O}(sj)$ storage locations. The computational work per iteration is bounded independent of j; it is $\mathcal{O}(n)$ flops in addition to the arithmetic work required for the evaluation of Av_j. In the special case when $s = 0$, Algorithm 4.4 simplifies to a minimal residual method for the solution of linear systems of equations with a symmetric, possibly indefinite, matrix.

We conclude this section with a comment on FOM, an iterative method that is closely related to GMRES; see Saad [15, section 6.4]. The jth iterate determined by

FOM, $x_j^{\mathrm{FOM}} \in x_0 + \mathcal{K}_j(A, r_0)$, satisfies

$$b - A\, x_j^{\mathrm{FOM}} \perp \mathcal{K}_j(A, r_0).$$

From, e.g., [15, section 6.5.5] we know that the iterate x_j^{FOM} exists if and only if $|s_j| = \|r_{j-1}\|/\|r_j\| < 1$, which is equivalent to $c_j \neq 0$, where s_j and c_j are entries of the Givens rotation Ω_{j+1}; see (4.4). In this case, the relation between x_j^{FOM} and the GMRES iterate x_j is given by

$$x_j = s_j^2\, x_{j-1} + (1 - s_j^2)\, x_j^{\mathrm{FOM}};$$

see Saad [15, section 6.5.5] for details. A comparison with (4.10) shows that

$$x_j^{\mathrm{FOM}} = \frac{\gamma_j}{c_j} z_{j+1},$$

i.e., the vectors z_{j+1} are FOM iterates up to normalization.

5. Computed examples. Linear systems of equations (1.2) with matrices of the form

$$A = \begin{bmatrix} A_{1,1} & A_{1,2} \\ A_{2,1} & A_{2,2} \end{bmatrix} \in \mathbb{R}^{n\times n},$$

with a symmetric leading principal submatrix $A_{1,1} \in \mathbb{R}^{(n-\ell)\times(n-\ell)}$ and $A_{1,2}, A_{2,1}^* \in \mathbb{R}^{(n-\ell)\times\ell}$, $A_{2,2} \in \mathbb{R}^{\ell\times\ell}$, arise in many applications. Example 5.1 outlines a path following method that gives rise to matrices of this kind, and Examples 5.2–5.4 discuss the solution of integral equations. All computations were carried out in MATLAB with machine epsilon about $2 \cdot 10^{-16}$.

Example 5.1. We are interested in computing the solution u of the nonlinear boundary value problem

$$-\Delta u - \lambda \exp(u) = 0 \quad \text{in} \quad S, \tag{5.1}$$

$$u = 0 \quad \text{on} \quad \partial S \tag{5.2}$$

as a function of the parameter λ, where Δ denotes the Laplacian, S the unit square, and ∂S its boundary. This problem is known as the Bratu problem and is a common test problem for path following methods. We discretize S by a uniform grid with $(\ell-1)^2$ interior grid points (s_k, t_k), where $t_k = s_k = k/\ell$, $1 \leq k < \ell$, and approximate the Laplacian by the standard five-point stencil. This yields a system of $(\ell-1)^2$ nonlinear equations

$$G(w, \lambda) = 0, \tag{5.3}$$

where the entries of the vector $w \in \mathbb{R}^{(\ell-1)^2}$ are approximations of the function u at the grid points. Numerous techniques for computing $w(\lambda)$ as λ is increased from, say, λ_0 to λ_1 are available; see, e.g., [1, 5, 6] and the references therein.

The matrix $\partial G/\partial w$ is singular at turning points (w, λ) of the path $\lambda \to (w(\lambda), \lambda)$, and one often introduces an auxiliary parameter η in order to be able to traverse these points. Thus, let $\lambda = \lambda(\eta)$ and assume that $w(\lambda(\hat{\eta}))$ is available, where

$\lambda_0 \leq \lambda(\hat{\eta}) \leq \lambda_1$. We would like to determine $\lambda(\hat{\eta}+\delta\eta)$ and $w(\lambda(\hat{\eta}+\delta\eta))$. Introduce the function

$$L(w,\lambda,\delta\eta) = d^*(w - w(\lambda(\hat{\eta}))) + c(\lambda - \lambda(\hat{\eta})) - \delta\eta \tag{5.4}$$

for some $d \in \mathbb{R}^{(\ell-1)^2}$ and $c \in \mathbb{R}$. The choice of d and c will be commented on below. Let $(w^{(j)}, \lambda^{(j)})$ be an available approximation of the solution of

$$\begin{aligned} G(w,\lambda) &= 0, \\ L(w,\lambda,\delta\eta) &= 0. \end{aligned} \tag{5.5}$$

Newton's method can be used to determine an improved approximation

$$(w^{(j+1)}, \lambda^{(j+1)}) = (w^{(j)} + \delta w, \lambda^{(j)} + \delta\lambda)$$

of the solution $(w(\lambda(\hat{\eta}+\delta\eta)), \lambda(\hat{\eta}+\delta\eta))$ of (5.5), where δw and $\delta\lambda$ satisfy

$$\begin{bmatrix} G_w^{(j)} & G_\lambda^{(j)} \\ d^* & c \end{bmatrix} \begin{bmatrix} \delta w \\ \delta\lambda \end{bmatrix} = \begin{bmatrix} -G^{(j)} \\ -L^{(j)} \end{bmatrix}, \tag{5.6}$$

with

$$\begin{aligned} G^{(j)} &= G(w^{(j)}, \lambda^{(j)}), & L^{(j)} &= L(w^{(j)}, \lambda^{(j)}, \delta\eta), \\ G_w^{(j)} &= \tfrac{\partial}{\partial w} G(w^{(j)}, \lambda^{(j)}), & G_\lambda^{(j)} &= \tfrac{\partial}{\partial \lambda} G(w^{(j)}, \lambda^{(j)}). \end{aligned}$$

The vector d should be chosen to make the matrix in (5.6) nonsingular even when G_w is singular. This allows simple turning points to be traversed. The parameter η is sometimes chosen to be arc length or pseudo–arc length of the curve $\lambda \to (w(\lambda), \lambda)$. The quantities d, c in (5.4) then may be defined by, e.g., $d = dw(\lambda(\hat{\eta}))/d\eta$, $c = d\lambda(\hat{\eta})/d\eta$.

To illustrate the performance of Algorithm 4.4, we discretize (5.1) on a uniform grid with $\ell = 26$. The matrix in (5.6) then is of size 626×626. We choose $\lambda = \exp(\eta) - 1$ and seek to determine the solution of (5.5) with $\delta\eta = 10$, starting with $w^{(0)} = 0$ and $\lambda^{(0)} = 0$, i.e., $x_0 = 0$ in Algorithm 4.4. Then $G_w^{(0)}$ is the negative discrete Laplacian, $G_\lambda^{(0)} = -[1,1,\ldots,1]^*$, $G^{(0)} = 0$, and $L^{(0)} = -\delta\eta$. We let $c = 1$ and, since $\partial w/\partial\eta$ is the largest at the center of the unit square, we choose $d = e_{(\ell-1)^2/2}$. This defines the matrix in (5.6), which we will refer to as A. It has skew-symmetric part of rank $s = 2$; cf. (1.1). We choose

$$f_1 = [1,1,\ldots,1,0]^* - e_{(\ell-1)^2/2}, \quad f_2 = e_{(\ell-1)^2+1}, \quad g_1 = f_2, \quad g_2 = -f_1$$

in the computations.

Algorithm 4.4 reduces the residual error from 10 ($= |\delta\eta|$) to $1.84 \cdot 10^{-7}$ in 50 iterations. In the present example, the numerical values of $\|b - Ax_{50}\|$, $\|r_{50}\|$ as computed by (4.10), and $|\gamma_{50}|$ agree to at least five significant digits. Solution of (5.6) by a direct method gave x_{direct} with $\|x_{\text{direct}} - x_{50}\| = 1.42 \cdot 10^{-10}$. Let x'_{50} denote the approximate solution determined by standard GMRES,[1] and let $r'_{50} = b - Ax'_{50}$. Then $\|r'_{50}\| = 1.84 \cdot 10^{-7}$, $\|x_{\text{direct}} - x'_{50}\| = 1.42 \cdot 10^{-10}$, and $\|x'_{50} - x_{50}\| = 4.90 \cdot 10^{-12}$.

[1]Standard GMRES refers to the commonly used GMRES implementation based on the Arnoldi process with orthogonalization of the Arnoldi vectors by the modified Gram–Schmidt method.

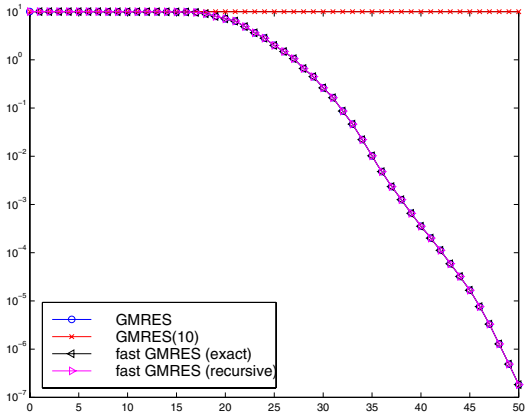


FIG. 5.1. *Residual norms for Algorithm* 4.4 *applied to the data of Example* 5.1. *For comparison, we show both the norm of the exact residuals* $\|b - Ax_k\|$ *(symbol* $\triangleleft$*) and the recursively computed residual norms* $|\gamma_k|$ *(symbol* $\triangleright$*), as well as the norm of the residuals* r'_k *(symbol* $\circ$*) obtained by standard GMRES, which are all of the same size. In contrast, restarted GMRES*(10) *(symbol* $\times$*) fails to converge.*

Figure 5.1 shows the residual errors for standard GMRES and Algorithm 4.4. Let x_k denote the iterates computed by Algorithm 4.4 and let γ_k be the recursively evaluated quantities in the algorithm, such that (in exact arithmetic) $|\gamma_k| = \|b - Ax_k\|$. Figure 5.1 displays $|\gamma_k|$, referred to as *fast GMRES (recursive)*, as well as the evaluated norms $\|b - Ax_k\|$, referred to as *fast GMRES (exact)*, for $0 \leq k \leq 50$. The $|\gamma_k|$ are seen to be accurate approximations of $\|b - Ax_k\|$. Moreover, the latter quantities are of the same size as the residual norms produced by standard GMRES.

Convergence is slow during the first 15 iterations and can be sped up by the use of a preconditioner. Note that a symmetric positive definite preconditioner would not change the rank of the skew-symmetric part.

Algorithm 4.4 requires about the same computer storage as GMRES restarted every $2s+6$ iterations. The latter method is referred to as restarted GMRES$(2s+6)$. We also compare Algorithm 4.4 to restarted GMRES$(2s+6)$. For the present example restarted GMRES$(2s+6)$ with $s = 2$ fails to converge; see Figure 5.1.

Both standard and restarted GMRES are implemented using modified Gram–Schmidt orthogonalization of the Arnoldi vectors. Algorithm 4.4 explicitly orthogonalizes each new Arnoldi vector v_{k+1} only against the two most recently generated vectors, v_k and v_{k-1}. Therefore, the orthogonality properties of the matrices $V_k = [v_1, v_2, \ldots, v_k]$ determined by standard GMRES and Algorithm 4.4 in finite precision arithmetic may differ. Figure 5.2 displays the quantities $\|I_k - V_k^* V_k\|^2$, for $1 \leq k \leq 50$, for matrices V_k determined by standard GMRES and Algorithm 4.4. In this example, the columns of the matrices V_k determined by Algorithm 4.4 are closer to orthonormal than those determined by standard GMRES.

Example 5.2. The integral equation

$$\gamma\, u(\alpha) + \frac{1}{\pi} \int_{-1}^{1} \frac{d}{d^2 + (\alpha - \beta)^2} u(\beta) d\beta = f(\alpha), \qquad -1 \leq \alpha \leq 1, \tag{5.7}$$

with $\gamma = 1$ and d a positive constant, is known as Love's integral equation. It arises in electrostatics; see, e.g., Baker [3, p. 258]. Let $f(\alpha) = (1 + \alpha)^{1/2}$, let $d = 1/10$, and discretize (5.7) by a Nyström method based on the composite trapezoidal rule with

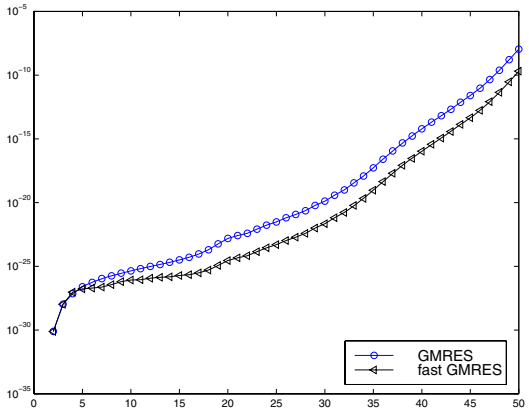


FIG. 5.2. *Orthonormality of the Arnoldi vectors for Example* 5.1: $\|I_k - V_k^* V_k\|^2$ *as a function of* k *for Algorithm* 4.4 *(symbol* $\triangleleft$*) and standard GMRES (symbol* $\circ$*).*

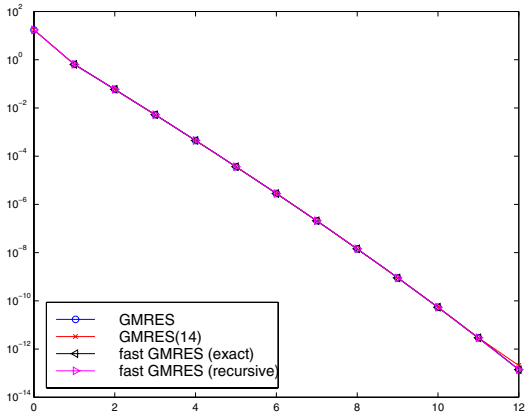


FIG. 5.3. *Residual norms for Algorithm* 4.4 *applied to the data of Example* 5.2. *For comparison, we show both the norm of the exact residuals* $\|b - Ax_k\|$ *(symbol* $\triangleleft$*) and the recursive residual norms* $|\gamma_k|$ *(symbol* $\triangleright$*), which are of the same size. The norm of the residuals* r'_k *obtained by standard GMRES (symbol* $\circ$*) and by restarted GMRES*(14) *(symbol* $\times$*) are also displayed.*

equidistant nodes $\alpha_k = \beta_k = (k-1)/(n-1)$, $1 \leq k \leq n$, $n = 300$. This gives a linear system of equations with a matrix of the form

$$A = \gamma I + KD, \tag{5.8}$$

where K is a symmetric Toeplitz matrix and $D = \mathrm{diag}[1/2, 1, 1, \ldots, 1, 1/2]$. The skew-symmetric part of A therefore is of rank $s = 4$. The memory requirement of Algorithm 4.4 is about the same as for restarted GMRES(14).

Figure 5.3 shows the residual errors for Algorithm 4.4 as given by $|\gamma_k|$ and $\|b - Ax_k\|$ for $0 \leq k \leq 12$, as well as the corresponding residual errors for standard GMRES. The initial approximate solution is $x_0 = 0$. The iterations are terminated as soon as the residual error for standard GMRES is of norm smaller than $1 \cdot 10^{-12}$. Convergence is rapid both for Algorithm 4.4 and standard GMRES, and the methods produce iterates with residual errors of nearly the same size.

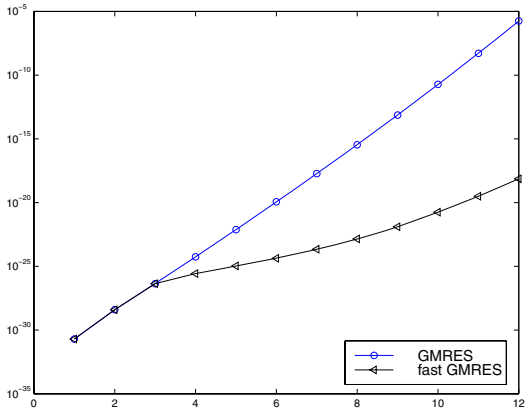


FIG. 5.4. *Orthonormality of the Arnoldi vectors for Example* 5.2: $\|I_k - V_k^* V_k\|^2$ *as a function of* k *for Algorithm* 4.4 *(symbol* $\triangleleft$*) and standard GMRES (symbol* $\circ$*).*

Figure 5.4 is analogous to Figure 5.2 and shows that the Arnoldi vectors generated by Algorithm 4.4 are slightly closer to being orthonormal than the Arnoldi vectors determined by standard GMRES.

The nonsymmetric matrix KD in (5.8) is the discretization of a compact integral operator. It has many eigenvalues close to the origin. Therefore the matrix (5.8) has many eigenvalues close to γ, which has the value one in Example 5.2. In the following examples, we will reduce γ. This reduces the rate of convergence and illustrates that, differently from Examples 5.1 and 5.2, the Arnoldi vectors determined by Algorithm 4.4 may be less close to orthonormal than the Arnoldi vectors determined by the Arnoldi process in the standard GMRES implementation.

Example 5.3. We modify the integral equation (5.7) of Example 5.2 by setting $\gamma = 0.1$. This change of γ reduces the rate of convergence. Discretization is carried out in the same manner as in Example 5.2. We use the same initial approximate solution and stopping criterion as in Example 5.2.

Figure 5.5 displays the norm of the residual errors for Algorithm 4.4, standard GMRES, and restarted GMRES(14) and is analogous to Figure 5.3. Figure 5.5 shows the residual errors r_{21} and r_{22} determined by Algorithm 4.4 to be of slightly larger norm than the corresponding residual errors determined by standard GMRES. The cause for this can be found in Figure 5.6(a), which shows the quantities $\|I_k - V_k^* V_k\|^2$ for $1 \leq k \leq 22$. The figure shows the Arnoldi vectors computed by Algorithm 4.4 to be slightly less close to orthonormal than are the Arnoldi vectors determined by standard GMRES.

Figure 5.6(b) displays $\|I_{m+1} - V_{m-k:k}^* V_{m-k:k}\|^2$ as a function of k for $m = 1, 2, \ldots, 5$, thus measuring the orthonormality between the last $m+1$ Arnoldi vectors computed by Algorithm 4.4. Orthonormality is lost fairly rapidly for $m \geq 3$.

Example 5.4. We modify the integral equation (5.7) of Examples 5.2 and 5.3 by setting $\gamma = 0.01$. This change of γ reduces the rate of convergence compared with Example 5.3. Discretization is carried out in the same manner as in Examples 5.2 and 5.3, and we use the same initial approximate solution and stopping criterion as in those examples.

Figure 5.7 displays the norm of the residual errors for Algorithm 4.4, standard GMRES, and restarted GMRES(14) and is analogous to Figure 5.5. Figure 5.7 shows

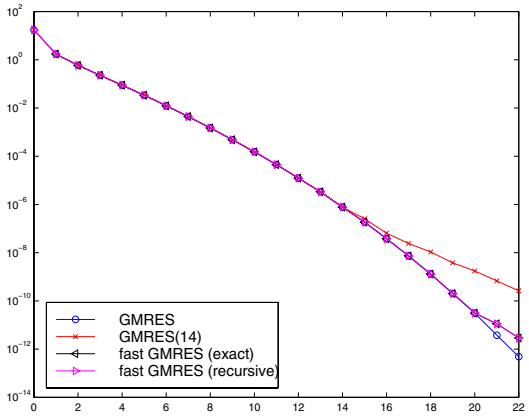
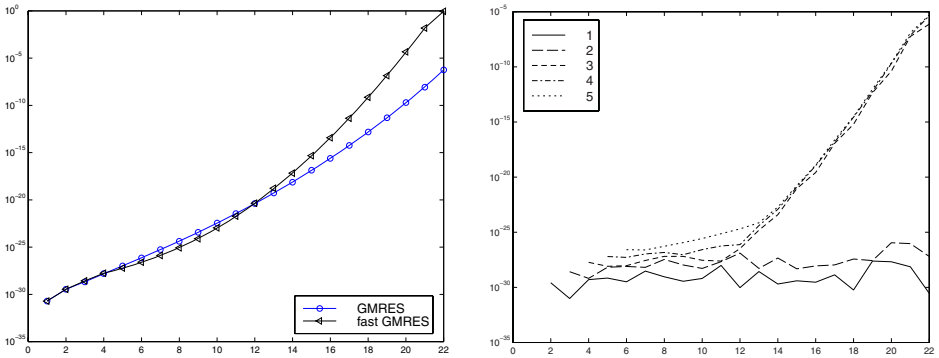


FIG. 5.5. *Residual norms for Algorithm* 4.4 *applied to the data of Example* 5.3. *For comparison, we display both the norm of the exact residuals* $\|b - Ax_k\|$ *(symbol* $\triangleleft$*) and the recursive residual norms* $|\gamma_k|$ *(symbol* $\triangleright$*), which are of the same size, and slightly smaller than those obtained for restarted GMRES*(14) *(symbol* $\times$*). The norms of the residuals* r'_k *determined by standard GMRES (symbol* $\circ$*) are somewhat smaller for* $k \geq 21$.



(a) (b)

FIG. 5.6. *Orthonormality of the Arnoldi vectors for Example* 5.3: (a) $\|I_k - V_k^* V_k\|^2$ *as a function of* k *for Algorithm* 4.4 *(symbol* $\triangleleft$*) and standard GMRES (symbol* $\circ$*).* (b) *From bottom to top,* $\|I_{m+1} - V_{m-k:k}^* V_{m-k:k}\|^2$ *as a function of* k *for* $m = 1, 2, \ldots, 5$ *for Algorithm* 4.4.

Algorithm 4.4 to reduce the norm of the residual error slower than standard GMRES, but faster than restarted GMRES(14).

The reason for the slower convergence of Algorithm 4.4 is the loss of orthonormality of the Arnoldi vectors generated by the algorithm. The latter is illustrated by Figures 5.8.

Examples 5.3 and 5.4 illustrate that the iterates determined by Algorithm 4.4 may converge slower to the solution than the iterates determined by standard GMRES. A reason for this appears to be that the Arnoldi vectors generated by Algorithm 4.4 may be far from orthonormal; see Example 5.4. The loss of orthogonality and its effect on the convergence of GMRES has received considerable attention in the literature; see, e.g., [8, 10, 13, 14, 16, 17]. For instance, Simoncini and Szyld [16] recently pointed out that loss of orthogonality does not prevent a near-optimal rate of convergence,

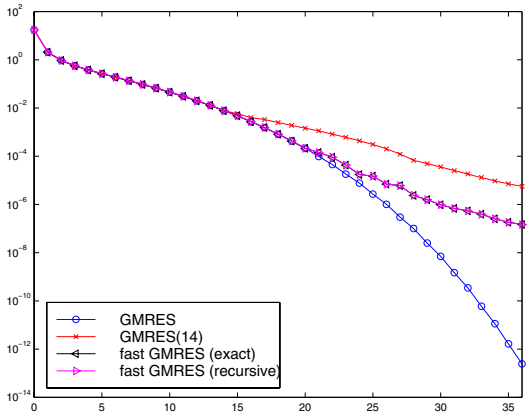


FIG. 5.7. *Residual norms for Algorithm* 4.4 *applied to the data of Example* 5.3. *For comparison, we show both the norm of the exact residuals* $\|b - Ax_k\|$ *(symbol* $\triangleleft$*) and the recursive residual norms* $|\gamma_k|$ *(symbol* $\triangleright$*), which are of the same size, and smaller than those obtained by restarted GMRES*(14) *(symbol* $\times$*). The norms of the residuals* r'_k *obtained by the standard GMRES (symbol* $\circ$*) are much smaller for* $k \geq 30$.

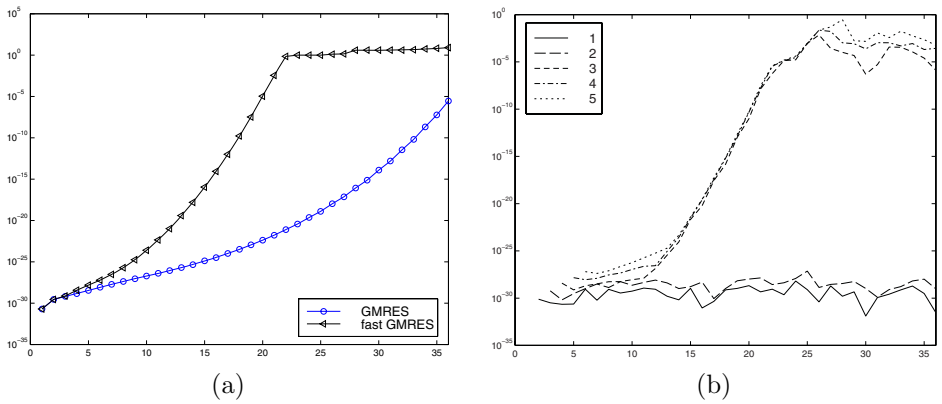


FIG. 5.8. *Orthonormality of the Arnoldi vectors for Example* 5.3: (a) $\|I_k - V_k^* V_k\|^2$ *as a function of* k *for Algorithm* 4.4 *(symbol* $\triangleleft$*) and standard GMRES (symbol* $\circ$*).* (b) *From bottom to top,* $\|I_{m+1} - V_{m-k:k}^* V_{m-k:k}\|^2$ *as a function of* k *for* $m = 1, 2, \ldots, 5$ *for Algorithm* 4.4.

provided that each new Arnoldi vector generated has a sufficiently large angle with the space spanned by the already available Arnoldi vectors. Example 5.4 suggests that the loss of orthogonality also may reduce this angle.

6. Conclusion. Linear systems of equations with a matrix that satisfies (1.1) with a small value of s arise in a variety of applications. For many, but not all, linear systems of equations of this kind, Algorithm 4.4 converges like standard GMRES, but requires less computer storage and arithmetic work. In all our experiments, Algorithm 4.4 converges faster than restarted GMRES($2s+6$), which demands roughly the same amount of computer storage as Algorithm 4.4.

Acknowledgment. We would like to thank a referee for comments.

REFERENCES

[1] E. L. Allgower and K. Georg, *Numerical Continuation Methods*, Springer, Berlin, 1990.
[2] G. Arnold, N. Cundy, J. van den Eshof, A. Frommer, S. Krieg, Th. Lippert, and K. Schäfer, *Numerical methods for the QCD overlap operator* II: *Optimal Krylov subspace methods*, in QCD and Numerical Analysis III, A. Boricci, A. Frommer, B. Joó, A. Kennedy, and B. Pendleton, eds., Lect. Notes Comput. Sci. Eng. 47, Springer, Berlin, 2005, pp. 153–167.
[3] C. T. H. Baker, *Numerical Treatment of Integral Equations*, Clarendon Press, Oxford, 1977.
[4] T. Barth and T. Manteuffel, *Multiple recursion conjugate gradient algorithms, part* I: *Sufficient conditions*, SIAM J. Matrix Anal. Appl., 21 (2000), pp. 768–796.
[5] D. Calvetti and L. Reichel, *Iterative methods for large continuation problems*, J. Comput. Appl. Math., 123 (2001), pp. 217–240.
[6] C.-S. Chen, N.-H. Lu, and Z.-L. Weng, *Conjugate gradient methods for continuation problems* II, J. Comput. Appl. Math., 62 (1995), pp. 197–216.
[7] P. Concus and G. H. Golub, *A generalized conjugate gradient method for nonsymmetric systems of linear equations*, in Computing Methods in Applied Science and Engineering, R. Glowinski and J. L. Lions, eds., Springer, New York, 1976, pp. 56–65.
[8] J. Drkosova, A. Greenbaum, M. Rozloznik, and Z. Strakos, *Numerical stability of GMRES*, BIT, 35 (1995), pp. 309–330.
[9] Yu. Eidelman, I. Gohberg, and V. Olshevsky, *The QR iteration method for Hermitian quasiseparable matrices of an arbitrary order*, Linear Algebra Appl., 404 (2005), pp. 305–324.
[10] A. Greenbaum, M. Rozloznik, and Z. Strakos, *Numerical behavior of the modified Gram-Schmidt GMRES implementation*, BIT, 37 (1997), pp. 706–719.
[11] C. Jagels and L. Reichel, *The isometric Arnoldi process and an application to iterative solution of large linear systems*, in Iterative Methods in Linear Algebra, R. Beauwens and P. de Groen, eds., Elsevier, Amsterdam, 1992, pp. 361–369.
[12] C. Jagels and L. Reichel, *A fast minimal residual algorithm for shifted unitary matrices*, Numer. Linear Algebra Appl., 1 (1994), pp. 555–570.
[13] C. C. Paige, M. Rozložník, and Z. Strakoš, *Modified Gram–Schmidt (MGS), least squares, and backward stability of MGS-GMRES*, SIAM J. Matrix Anal. Appl., 28 (2006), pp. 264–284.
[14] C. C. Paige and Z. Strakoš, *Residual and backward error bounds in minimum residual Krylov subspace methods*, SIAM J. Sci. Comput., 23 (2002), pp. 1898–1923.
[15] Y. Saad, *Iterative Methods for Sparse Linear Systems*, 2nd ed., SIAM, Philadelphia, 2003.
[16] V. Simoncini and D. B. Szyld, *The effect of non-optimal bases on the convergence of Krylov subspace methods*, Numer. Math., 100 (2005), pp. 711–733.
[17] Z. Strakoš and J. Liesen, *On numerical stability in large scale linear algebraic computations*, Z. Angew. Math. Mech., 85 (2005), pp. 307–325.
[18] O. Widlund, *A Lanczos method for a class of nonsymmetric systems of linear equations*, SIAM J. Numer. Anal., 15 (1978), pp. 801–812.