

LEAST SQUARES POLYNOMIALS IN THE COMPLEX PLANE AND THEIR USE FOR SOLVING NONSYMMETRIC LINEAR SYSTEMS*

YOUSSEF SAAD†

Abstract. We propose an algorithm for computing a class of least squares polynomials on polygonal regions of the complex plane. An important application of this technique to solving large sparse linear systems is considered. The advantage of using general polygonal regions instead of ellipses as was done in previous work, is that elliptic regions may fail to accurately represent the convex hull of the spectrum of the matrix A . An attractive feature of the algorithm is that it does not explicitly require any numerical integration. Numerical experiments show that the least-squares based methods for solving linear systems are competitive with the Chebyshev based methods and are more reliable.

Key words. least squares polynomials, nonsymmetric linear systems, semi-iterative method, hybrid methods

AMS(MOS) subject classification. 65F10

1. Introduction. Consider the linear system of equations

$$(1) \quad Ax = f,$$

where A is an arbitrary real matrix of size N . Note that (1) can be the result of a preconditioning technique applied to another linear system. Many iterative methods for solving (1) amount to the polynomial iteration

$$(2) \quad x_n = x_0 + s_n(A)r_0$$

where x_0 is some initial approximation to the solution, $r_0 = f - Ax_0$ and s_n is a polynomial of degree $n - 1$. The residual vector $r_n = f - Ax_n$ is such that

$$(3) \quad r_n = [I - As_n(A)]r_0 \equiv R_n(A)r_0$$

where $R_n(\lambda) = 1 - \lambda s_n(\lambda)$ is a polynomial of degree n , known as the residual polynomial, which satisfies the constraint

$$(4) \quad R_n(0) = 1.$$

Clearly, one wants to find a polynomial s_n so that $\|R_n(A)r_0\|$ is as small as possible, where $\|\cdot\|$ represents the Euclidean norm. Several methods are explicitly based on this formulation, in that they attempt to compute the approximation x_n for which the Euclidean norm $\|r_n\|$ is minimal, see, e.g., [2], [5], [14]. However, the corresponding algorithms do not always constitute a good choice, especially in the following situations:

- when A is not positive real in which case some of these techniques may fail, see [2];
- in time-dependent or parameter-dependent problems where the conjugate gradient techniques do not take advantage of the iteration parameters that have been computed during the previous time steps;
- in some modern architecture machines, because the conjugate gradient type methods are highly sequential in nature.

* Received by the editors July 19, 1983; accepted for publication (in revised form) September 6, 1985. This work was supported in part by the U.S. Office of Naval Research under grant N000014-76-C-0277, in part by the Department of Energy under grant AC02-81ER10996, and in part by the U.S. Air Force under grant AFOSR-83-0097.

† Department of Computer Science, Yale University, New Haven, Connecticut 06520.

Assume that A is diagonalizable, and denote by $\sigma(A) = \{\lambda_i\}_{i=1,N}$ its spectrum. Expanding the initial residual vector r_0 in the basis of eigenvectors $\{u_i\}_{i=1,N}$ as

$$r_0 = \sum_{i=1}^N \xi_i u_i,$$

we obtain the following expansion for the vector r_n :

$$r_n = \sum_{i=1}^N R_n(\lambda_i) \xi_i u_i.$$

Then it is clear that instead of minimizing the residual norm $\|R_n(A)r_0\|$, one is tempted to minimize the discrete uniform norm

$$(5) \quad \max_{\lambda \in \sigma(A)} |R_n(\lambda)|,$$

over all polynomials R_n of degree n satisfying the constraint (4). Clearly, the eigenvalues λ_i of A are usually not known so one replaces the spectrum $\sigma(A)$ by some region H of the complex plane that includes $\sigma(A)$, and R_n is then chosen to minimize

$$(6) \quad \max_{\lambda \in H} |R_n(\lambda)|,$$

over all polynomials of degree n so that $R_n(0) = 1$. Such polynomials will be referred to as the minimax polynomials. A well-known method based on this approach is the Chebyshev iteration method studied by Wrigley [19], Manteuffel [6]–[8] and others. There, H is taken to be an ellipse with center c and focal distance d , which contains the convex hull of $\sigma(A)$. If the origin is outside the ellipse, the minimax polynomial reduces to the scaled and shifted Chebyshev polynomial

$$(7) \quad R_n(\lambda) = T_n\left(\frac{c-\lambda}{d}\right) / T_n\left(\frac{c}{d}\right).$$

When d and c are real, the polynomial (7) is known to minimize the uniform norm on an ellipse centered at c and with focal distance d , over all polynomials satisfying the constraint (4). The three term recurrence of the Chebyshev polynomial induces an elegant algorithm for generating the approximation x_n that uses only three vectors of storage. An adaptive algorithm that obtains the optimal ellipse containing the convex hull of the eigenvalues of A was proposed by Manteuffel [8], and was recently improved by Elman et al. [3].

There are, however, a few drawbacks to the Chebyshev iteration. The most serious drawback is that the computed convex hull may have eigenvalues on both sides of the imaginary axis of the complex plane as may occur when A is not positive real, i.e., when its symmetric part $(A + A^T)/2$ is not positive definite. This situation is not too uncommon, especially for preconditioned systems [2]. Since there is no ellipse containing the computed convex hull and excluding the origin the method breaks down. Clearly, the computed spectrum can still be enclosed in two convex regions of the complex plane each on one side of the imaginary axis and the difficulty can be resolved by computing a polynomial which satisfies the constraint $R_n(0) = 1$, and which is “small” in the union of the two regions in some sense.

A second drawback is that the optimal ellipse which encloses the spectrum often does not accurately represent the spectrum, which may result in slow convergence. Typical examples reported in [17] are those of a boomerang-shaped spectrum and a cross-shaped spectrum in the complex plane.

To overcome these two drawbacks, Smolarski and Saylor [17] proposed to use polygonal regions having a relatively small number of edges instead of ellipses. Since

the spectrum is discrete, it is always possible to enclose it in a set H consisting of one or more polygonal regions. The problem is then to find a polynomial s_n such that $|R_n(\lambda)|$ is small inside and on the boundary of H . By the maximum principle, the maximum modulus of $|1 - \lambda s_n(\lambda)|$ is achieved on the boundary and therefore it is sufficient to regard the problem as being defined on the boundary. Instead of computing the best uniform polynomial on the polygon, Smolarski and Saylor suggest the use of the least squares residual polynomial, i.e., the polynomial R_n satisfying the constraint $R_n(0) = 1$, and minimizing the L_2 -norm $\|1 - \lambda s_n(\lambda)\|_w$ with respect to some weight $w(\lambda)$ on the boundary of H .

The method proposed in [17] for computing the least squares polynomial s is based on classical moments and on the use of the Kernel polynomial formulation of the least squares polynomial due to Stiefel [18]. Such a process is unstable as was noted by Smolarski and Saylor. The reason for this instability is that the moment matrix associated with the powers λ^i is highly ill-conditioned. In the context of computing orthogonal polynomials on *real intervals*, it is well known that a better approach is to use the modified moments $\langle t_i(\lambda), t_j(\lambda) \rangle$, where $\{t_j\}$ is some suitable basis of polynomials. This is the foundation of the well-known modified moment method for computing orthogonal polynomials in the real case [4], [15]. In this paper an algorithm using explicitly the modified moment matrix will be developed for the problem of computing least squares polynomials in the complex plane, which satisfy the constraint $R_n(0) = 1$. We will see that a reasonable basis $\{t_j\}$ is the basis of Chebyshev polynomials suitably shifted and scaled. The polynomial s_n is directly expressed in the stable basis $\{t_j\}$ instead of the basis of the successive powers as in [16], [17]. Moreover, our algorithm avoids computing roots of the residual polynomial and using Richardson's iteration. Furthermore, we will show that numerical integration is not required, thanks to a technique adapted from [13].

Section 2 presents an algorithm based on the modified moments for computing the least squares polynomial for solving linear systems. In § 3 a hybrid method similar to the one presented in [3] is outlined. Finally, § 4 describes a few numerical experiments.

2. Computation of the least squares polynomial. When computing orthogonal polynomials p_0, p_1, \dots, p_n , or least squares polynomials in the basis $\{1, \lambda, \lambda^2, \dots\}$ one needs to factor the Gram matrix $\{\langle \lambda^{i-1}, \lambda^{j-1} \rangle\}$ often referred to as the moment matrix [16], [17]. This matrix can become very ill-conditioned and any method based on this approach will be limited to polynomials of low degree, e.g., not exceeding 10 [17]. A more reliable alternative is to replace the basis $\{\lambda^{i-1}\}$ by a more stable basis of polynomials $\{t_j(\lambda)\}$, provided for example by Chebyshev polynomials. The moment matrix is then replaced by the new Gram matrix $M_n = \{\langle t_i, t_j \rangle\}_{i,j=0,n}$ which will be referred to as the modified moment matrix. In this section we will formulate a method that uses the modified moment matrix to solve the least squares problem introduced in the previous section. In order to define the problem properly and provide the means for solving it, we will answer the following questions in turn:

1. How to compute the convex hull H and define a weight function $w(\lambda)$ on the boundary of H ?

2. How to choose a “stable” basis of polynomials $\{t_j(\lambda)\}_{j=0,1,\dots,n}$?

3. How to compute the modified moment matrix $M_n = \{\langle t_i, t_j \rangle\}_{i,j=0,1,\dots,n}$?

4. How to get the least squares polynomial in the basis $\{t_j\}$?

2.1. The convex hull and the weight function. To obtain the convex hull of the spectrum of A , we need eigenvalue estimates. There are various ways of obtaining

estimates of eigenvalues of A , but it is clear that these estimates will be obtained as the result of an adaptive method similar to those described in [3] and [8]. Let D be a set of estimates to the eigenvalues of A provided by any such process. There are many ways of obtaining the convex hull H of D . We will not describe any particular one of them but we should point out that since we deal with real matrices the spectrum is symmetric with respect to the real axis and we will only need the upper half part H^+ of the convex hull H . Suppose that the $\mu+1$ points h_0, h_1, \dots, h_μ constitute the vertices of the upper part H^+ of the convex hull H of D . Clearly, the convex hull H is obtained from H^+ by symmetry, i.e., by adding the points: $h_{\mu+k} = \bar{h}_{\mu-k}$, $k = 1, \dots, \mu$; see Fig. 2.1. Note that h_0 and h_μ are real and that $h_{2\mu} = h_0$.

If the convex hull H contains the origin, then the residual polynomial $R_n(\lambda)$ will be a poor polynomial. Indeed, by the maximum principle $R_n(\lambda)$ will reach its maximum modulus on the boundary ∂H and since $R_n(0) = 1$ this maximum will be no less than one. In this situation we must redefine H to be the union of the convex hull H_1 of the eigenvalues that have negative real parts and the convex hull H_2 of the eigenvalues with positive real parts. This set consisting of the two convex regions H_1 and H_2 will still be referred to as the convex hull by abuse of language.

The problem we would like to solve is to minimize some least squares norm of the polynomial $1 - \lambda s_n(\lambda)$ over all polynomials s of degree $n-1$, so we must define

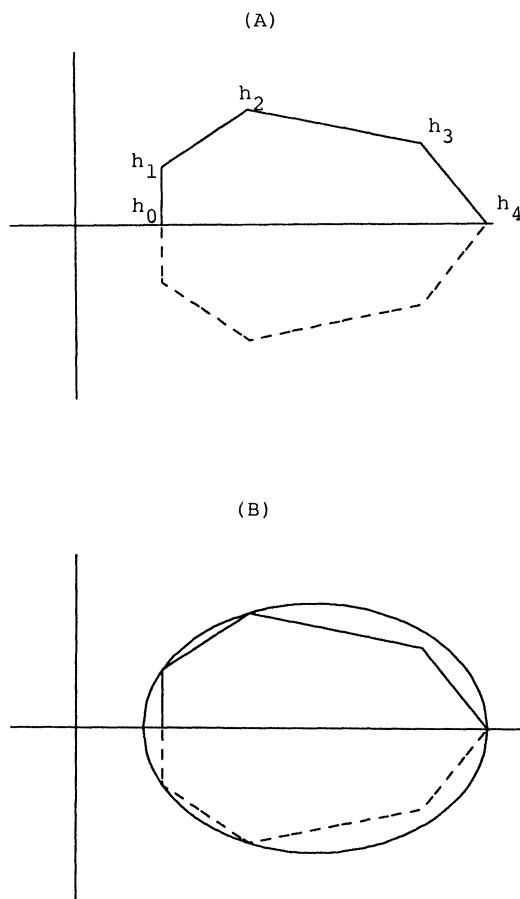


FIG. 2.1. The convex hull (A) and the ellipse of smallest area containing it (B).

an inner product on the boundary of the convex hull which will induce that norm. On each edge E_ν of the convex hull, $\nu = 1, \dots, \mu$, we must therefore choose a weight function $w_\nu(\lambda)$. Denoting by c_ν the center of the ν th edge and by d_ν its half width, i.e.,

$$(8) \quad c_\nu = \frac{1}{2}(h_\nu + h_{\nu-1}), \quad d_\nu = \frac{1}{2}(h_\nu - h_{\nu-1}),$$

we will consider on each edge the weight w_ν defined by

$$(9) \quad w_\nu(\lambda) = \frac{2}{\pi} |d_\nu^2 - (\lambda - c_\nu)^2|^{-1/2}.$$

This is nothing but a generalized Chebyshev weight, on each edge. The main reason for the choice of the above weight is simplicity as it will lead to Chebyshev polynomials. In fact we can use any generalization of a classical weight function on an interval $[-1, 1]$, for which the orthogonal polynomials are explicitly known.

We define $w(\lambda)$ as the function on the boundary of H whose restriction to each edge E_ν is $w_\nu(\lambda)$. The inner product on the space of complex polynomials \mathbf{P}_n of degree not exceeding n is therefore defined by

$$(10) \quad \langle p, q \rangle = \int_{\partial H} p(\lambda) \overline{q(\lambda)} w(\lambda) |d\lambda| = \sum_{\nu=1}^{\mu} \int_{E_\nu} p(\lambda) \overline{q(\lambda)} w_\nu(\lambda) |d\lambda|.$$

An important remark is that if p and q have real coefficients then we need only compute the integrals over the edges of the upper part of H because in this case

$$(11) \quad \langle p, q \rangle = 2 \operatorname{Re} \left\{ \int_{\partial H^+} p(\lambda) \overline{q(\lambda)} w(\lambda) |d\lambda| \right\},$$

where ∂H^+ denotes the upper part of the boundary ∂H of H .

2.2. The basis of Chebyshev polynomials. In approximation theory, it is well known that, in general, the use of the power basis $\{1, \lambda, \lambda^2, \dots, \lambda^{n-1}\}$ is to be avoided for stability reasons. Instead, if one is for example interested in approximating a given function in the interval $[-1, +1]$, a better alternative is to use the Chebyshev basis $\{T_j(\lambda)\}_{j=0,n}$. If the interval is $[\alpha, \beta]$ then a good basis is $\{T_j[(\lambda - c)/d]\}$ where $c = (\beta + \alpha)/2$, $d = (\beta - \alpha)/2$. By analogy, assume that the convex hull H can be enclosed in an ellipse centered at c , and having focal distance d and major semi axis a . Then a natural basis of polynomials is the following

$$(12) \quad t_j(\lambda) = T_j\left(\frac{\lambda - c}{d}\right) / T_j\left(\frac{a}{d}\right), \quad j = 0, 1, \dots, n.$$

We have normalized the polynomial so that its maximum modulus on the ellipse is one; see [10]. The stability of a given basis can be measured by the condition number of the corresponding Gram matrix. If the Gram matrix is highly ill-conditioned it will be difficult to compute the least squares polynomials in the corresponding basis. As a comparison, near-linear dependence of a system of vectors is often measured by the ratio of the largest to the smallest singular values of the system, i.e., by the square root of the condition number of its Gram matrix.

In what follows we would like to analyse the growth of the condition number of the modified moment matrix M_n of the system (12) as n increases. Let the boundary ∂H of the convex hull H consist in m edges, where m is not necessarily even as in the previous section. We will denote by $\mathcal{E}(c, d, a)$ the ellipse with center c , focal distance d and major semi axis a . In what follows, the condition number of a symmetric matrix refers to its spectral condition number, i.e., the ratio of its largest eigenvalue to its smallest eigenvalue. We can prove the following result [10].

PROPOSITION 1. Assume that the convex hull H is enclosed by the ellipse $\mathcal{E}(c, d, a)$ and that the boundary ∂H encloses the ellipse $\mathcal{E}(c, d_I, a_I)$ with $d_I \leq d$. Then the condition number $\tau(M_n)$ of the modified moment matrix M_n satisfies the inequality

$$(13) \quad \tau(M_n) \leq 2m(n+1)^2 \left(\frac{a + \sqrt{a^2 - d_I^2}}{a_I + \sqrt{a_I^2 - d_I^2}} \right)^{2n}.$$

The proof of this result along with a more detailed analysis of the condition numbers of the modified moment matrices can be found in [10]. A similar formula can be shown for the case when the condition $d_I \leq d$ is not satisfied, and also for the case where the two ellipses have different centers. In what follows we will refer to the coefficient which is elevated to the power $2n$ in (13) as the growth factor.

It is not known whether this upper bound is sharp. The result shows only that if H is well approximated by an ellipse, i.e., when a is close to a_I , then the condition number of M_n will not increase too rapidly. It strongly suggests that the ellipse $\mathcal{E}(c, d, a)$ must be the ellipse closest to ∂H in some sense, in order to have a small growth factor. We will discuss the choice of a good enclosing ellipse in more detail shortly. To illustrate the result, suppose for example that H is a rectangle centered at c , on the real axis, having half-length L on the real axis, and half-width l . Let us take the ellipse $\mathcal{E}(c, d, L)$ with semi-major axes L and l as the inner ellipse, where $d_I^2 = L^2 - l^2$ and the ellipse $\mathcal{E}(c, d_I, a)$ passing by the points $(c \pm L) + il$ as the outer ellipse. It is then easy to show that the growth factor becomes $(\sqrt{L} + \sqrt{l})/(\sqrt{L} - \sqrt{l})$. This is close to one when $l \ll L$. For $l = 0.1$, $L = 1$, and $n = 10$ we get $\tau(M_n) \leq 9.1 \times 10^4$. For $n = 20$ the bound becomes $\tau(M_n) \leq 3.1 \times 10^7$. These are only upper bounds and we can expect the actual condition numbers to be much smaller in general.

Another consequence of (13) is that it indicates that the condition number is likely to increase geometrically with n and that some caution must be taken in order to avoid a too high degree n . In practice this is relatively easy to achieve by performing the Choleski factorization in a progressive way. As will be seen in the next section, the moment matrix M_n can be built column by column. As column $j+1$ becomes available we can use the Choleski factorization of the moment matrix of size j , to get the Choleski factorization of that of size $j+1$. This is often referred to as a frontal method. The relative size of the diagonal elements encountered during the factorization gives an indication of the condition of the corresponding modified moment matrix. As soon as a diagonal element $l_{j+1,j+1}$ is considered too small we stop the process and work with the $j \times j$ modified moment matrix M_j instead of M_n , i.e. we work with polynomials of degree $j-1$ instead of n . Note that the cost of the Choleski factorization of the modified moment matrix is a negligible amount of the total cost of solving large linear systems. One can even afford to actually compute the inverses of the triangular factors at each step in order to get a good estimate for the condition number of the matrix M_n in order to stop at the appropriate degree.

We now address the question of selecting good parameters c and d . A simple idea is to use the parameters c and d of the ellipse of *smallest area* that contains H because we would like the ellipse to fit the convex hull as closely as possible, as is suggested by the above discussion. We restrict c to be real and d to be either real or purely imaginary. This means that the ellipse will be centered on the real axis and that its main axis is either along the real axis or the imaginary axis. It can be easily shown by induction that t_n has real coefficients even when d is purely imaginary, see [11]. Chebyshev polynomials have the additional advantage of satisfying a three term recurrence which facilitates the computation of $s_n(A)v$. Manteuffel has proposed an algorithm for computing the parameters of the ellipse that maximizes some complicated

convergence ratio [6]. Such an ellipse is different from the ellipse of smallest area containing H , which is far easier to determine. An algorithm for determining such an ellipse is described in [9].

2.3. Computing the modified moment matrix. A critical part of the computation of the least squares polynomials lies in the computation of the $(n+1) \times (n+1)$ Gram matrix M_n whose elements $m_{i,j}$ are defined by

$$(14) \quad m_{i,j} = \langle t_{j-1}, t_{i-1} \rangle, \quad i, j = 1, 2, \dots, n+1$$

and of its Choleski factorization. Note that from (11) and the fact that t_n has real coefficients, the coefficients $m_{i,j}$ are all real. Therefore, M_n is a symmetric positive definite real matrix.

Normally, the computation of (14) requires numerical integration but, as will be seen, this can be avoided by resorting to an idea developed in [13] in another context. We should point out that although we use a Chebyshev basis, the matrix M_n is still likely to become increasingly ill-conditioned as its size $n+1$ increases. Any error in the numerical integration may therefore be amplified, which could result in an inaccurate optimal polynomial.

Proceeding as in [13] we express the polynomials $t_j(\lambda)$ in terms of the Chebyshev polynomials

$$(15) \quad T_i\left(\frac{\lambda - c_\nu}{d_\nu}\right) = T_i(\xi), \quad i = 0, 1, \dots, j,$$

for each of the m edges E_ν , $\nu = 1, \dots, \mu$. Notice that the variable ξ is real for λ belonging to the edge E_ν . In other words, for each ν , $\nu = 1, 2, \dots, \mu$, we express each $t_j(\lambda)$, $j = 0, 1, \dots, n, \dots$, as

$$(16) \quad t_j(\lambda) = \sum_{i=0}^j \gamma_{i,j}^{(\nu)} T_i(\xi),$$

$$(17) \quad \text{where } \xi = \frac{\lambda - c_\nu}{d_\nu} \text{ is real.}$$

Each polynomial t_j will have μ expressions of this type, one for each edge E_ν , $i = 1, \dots, \mu$. Clearly, these μ expressions are redundant in that it is possible to obtain $\mu - 1$ of them from a single one, e.g., from the first. However, the process of changing bases can be unstable and will not be considered.

In order to build the moment matrix M_n , we need to be able to compute the expansion coefficients $\gamma_{i,j}^{(\nu)}$ of (16). This can be done thanks to the three term recurrence of the polynomials (12), which we write in the form

$$(18) \quad \beta_{k+1} t_{k+1}(\lambda) = (\lambda - \alpha_k) t_k(\lambda) - \delta_k t_{k-1}(\lambda).$$

Using the defining expressions (16) and (17) we obtain for each edge E_ν

$$\beta_{k+1} t_{k+1}(\lambda) = (d_\nu \xi + c_\nu - \alpha_k) \sum_{i=0}^k \gamma_{i,k}^{(\nu)} T_i(\xi) - \delta_k \sum_{i=0}^{k-1} \gamma_{i,k-1}^{(\nu)} T_i(\xi),$$

which provides the expression for t_{k+1} from those of t_k and t_{k-1} by noticing that

$$\xi T_i(\xi) = \frac{1}{2} [T_{i+1}(\xi) + T_{i-1}(\xi)], \quad i > 0, \quad \xi T_0(\xi) = T_1(\xi).$$

Thus, we have proved the following proposition.

PROPOSITION 2. *For each ν , $\nu = 1, 2, \dots, \mu$, the expansion coefficients $\gamma_{i,k}^{(\nu)}$ satisfy the following recurrence relation with respect to k :*

$$\beta_{k+1} \gamma_{i,k+1}^{(\nu)} = \frac{d_\nu}{2} [\gamma_{i+1,k}^{(\nu)} + \gamma_{i-1,k}^{(\nu)}] + (c_\nu - \alpha_i) \gamma_{i,k}^{(\nu)} - \delta_k \gamma_{i,k-1}^{(\nu)}, \quad i = 0, 1, \dots, k+1$$

with the notational conventions

$$\gamma_{-1,k}^{(\nu)} = \gamma_{1,k}^{(\nu)}, \quad \gamma_{i,k}^{(\nu)} = 0 \quad \text{for } i > k.$$

Once the coefficients $\gamma_{i,k}^{(\nu)}$ of (16), have been computed with the help of the above proposition, one can then compute the modified moment matrix by using the following result.

PROPOSITION 3. *Assuming the expressions (16) for t_j , $j = 0, 1, \dots, n$, the coefficients of the modified moment matrix M_n are given by*

$$m_{i+1,j+1} = 2 \operatorname{Re} \left\{ \sum_{\nu=1}^{\mu} \sum_{k=0}^i \gamma_{k,j}^{(\nu)} \bar{\gamma}_{k,i}^{(\nu)} \right\}, \quad i = 0, 1, \dots, j$$

where $\sum_{k=0}^i$ is defined by

$$\sum'_{k=0}^i \alpha_k \equiv 2\alpha_0 + \sum_{k=1}^i \alpha_k.$$

Proof. Follows from the remark (11), the change of variables (17) and the orthogonality of the (regular) Chebyshev polynomials. \square

The computation of an $n+1$ by $n+1$ Gram matrix M_n using the above two propositions requires $O(\mu n^3/3)$ multiplications. Remember, however, that this cost is not too significant in comparison with that involved in the linear system as n and μ will usually be much less than the dimension N of the matrix. In a typical code the upper limit for μ could be, for example, eight while the upper limit for n could be set to 40.

2.4. Getting the least squares polynomial s_n in the Chebyshev basis. Once the modified moment matrix M_n is available, our next task is to solve the least squares problem

$$(19) \quad \min_{s \in P_{n-1}} \|1 - \lambda s_n(\lambda)\|_w.$$

For the purpose of stability, the solution s_n will be directly expressed in the basis (16), i.e.,

$$(20) \quad s_n(\lambda) = \sum_{i=0}^{n-1} \eta_i t_i(\lambda).$$

Therefore, the problem is to find $\eta = (\eta_0, \eta_1, \dots, \eta_{n-1})^T$ so that

$$(21) \quad J(\eta) \equiv \|1 - \lambda s_n(\lambda)\|_w$$

is minimum. This optimization problem will be solved by translating it into a least squares *matrix problem* of dimension $n+1$.

Denoting by (\cdot, \cdot) the complex inner product in C^{n+1} , and letting

$$p(\lambda) = \sum_{i=0}^n \eta_i t_i(\lambda) \quad \text{and} \quad q(\lambda) = \sum_{i=0}^n \theta_i t_i(\lambda)$$

be any two polynomials of degree not exceeding n , it is clear that

$$(22) \quad \langle p, q \rangle = (M_n \eta, \theta)$$

where $\eta = (\eta_0, \eta_1, \dots, \eta_n)^T$ and $\theta = (\theta_0, \theta_1, \dots, \theta_n)^T$.

Consider now the n th degree polynomial $\lambda s_n(\lambda)$ which satisfies

$$\lambda s_n(\lambda) = \sum_{i=0}^{n-1} \eta_i \lambda t_i(\lambda) = \sum_{i=0}^{n-1} \eta_i [\beta_{i+1} t_{i+1}(\lambda) + \alpha_i t_i(\lambda) + \delta_i t_{i-1}(\lambda)].$$

This equality can be translated as follows: if $\eta = (\eta_0, \eta_1, \dots, \eta_{n-1})^T$ represents the vector of the coefficients of the polynomial s_n in the basis t_i , $i = 0, \dots, n-1$, then the polynomial $\lambda s_n(\lambda)$ is represented in the basis t_i , $i = 0, 1, \dots, n$ by the vector $T_n \eta$ where T_n is the $n+1$ by n tridiagonal matrix

$$T_n = \begin{bmatrix} \alpha_0 & \delta_1 & & & & \\ \beta_1 & \alpha_1 & \delta_2 & & & \\ & \beta_2 & \alpha_2 & \ddots & & \\ & & \ddots & \ddots & & \\ & & & \ddots & \ddots & \delta_{n-1} \\ & & & & \beta_{n-1} & \alpha_{n-1} \\ & & & & & \beta_n \end{bmatrix}.$$

As a consequence the polynomial $1 - \lambda s_n(\lambda)$ has coefficients $e_1 - T_n \eta$ in the basis $\{t_i\}_{i=0,n}$, where $e_1 = (1, 0, 0, \dots, 0)^T$. Using (22), we have

$$J(\eta)^2 = [e_1 - T_n \eta]^H M_n [e_1 - T_n \eta],$$

where x^H represents the transpose of the conjugate of x . Let

$$(23) \quad M_n = LL^T$$

be the Choleski factorization of M_n . Then

$$J(\eta) = \|L^T[e_1 - T_n \eta]\|.$$

Noticing that $L^T e_1 = l_{11} e_1 \equiv (l_{11}, 0, 0, \dots, 0)^T$ and denoting by F_n the $n+1$ by n upper Hessenberg matrix $F_n = L^T T_n$, we get the following expression for $J(\eta)$:

$$(24) \quad J(\eta) = \|l_{11} e_1 - F_n \eta\|.$$

The minimizer η^* of (24) is best computed by reducing the Hessenberg matrix F_n to upper triangular form using plane rotations. Note that a similar problem occurs in the GMRES method [14].

An alternative technique for solving the least squares problem (19) uses the Kernel polynomial formulation for getting $R_n(\lambda)$ and then obtains the approximate solution from the Richardson iteration after computing the roots of R_n [16]–[18]. Indeed, the polynomial R_n is known to satisfy the formula:

$$R_n(\lambda) = \left[\sum_{i=0}^n \bar{p}_i(0) p_i(\lambda) \right] / \left[\sum_{i=0}^n |p_i(0)|^2 \right]$$

where p_i , $i = 0, \dots, n$ is the sequence of orthogonal polynomials with respect to the weight w . By this method we would have first to compute the orthogonal polynomials, using the Choleski factorization of the modified moment matrix, and then get R_n . Once R_n is obtained, its roots must be calculated to compute x_n by the Richardson iteration. This last part of the process is not numerically reliable.

3. A hybrid method for solving linear systems of equations. As was already pointed out, a method based on the polynomial iteration (2) alone has a limited practical value,

because the eigenvalues of the matrix of A are not known beforehand. A common approach to deal with this difficulty is to combine the polynomial iteration with an adaptive scheme which will obtain eigenvalue estimates. Manteuffel's adaptive Chebyshev algorithm constitutes such a process. Another method, devised by Elman, Saad and Saylor [3], combines Chebyshev iteration with the GMRES method of [14] which is used to simultaneously compute eigenvalue estimates and improve the current approximation to the solution. We propose to replace the Chebyshev iteration of the hybrid method of [3] with the new polynomial iteration (2), where s_n is the least squares polynomial, while retaining the same adaptive step based on GMRES.

Let x_* be an approximation to the solution obtained from a certain number of iterations of the form (2), and let v_1 be the normalized residual $v_1 = r_* / \|r_*\|$, with $r_* = f - Ax_*$. The adaptive step starts by generating an orthonormal basis of the m -dimensional Krylov subspace $\text{span}\{v_1, Av_1, \dots, A^{m-1}v_1\}$ by an iteration of the form:

$$h_{j+1,j}v_{j+1} = Av_j - \sum_{i=1}^j h_{ij}v_i$$

known as Arnoldi's method [1], [12]. The orthonormal matrix $V_m \equiv [v_1, v_2, \dots, v_m]$ and the upper Hessenberg matrix $H_m \equiv [h_{ij}]$ are such that $V_m^T A V_m = H_m$. The eigenvalues of H_m are known to provide estimates for the outmost eigenvalues of A [12].

An important feature of the hybrid technique is that the information from Arnoldi's method can be taken advantage of for improving the approximation x_* . Indeed, one can compute the vector \tilde{x} belonging to the affine space $x_* + K_m$ which achieves the minimum of the residual norm $\|f - A\tilde{x}\|$. Let \bar{H}_m denote the $m+1$ by m matrix obtained by appending to H_m a row with single nonzero entry $h_{m+1,m}$ in position $m+1, m$. Then it can easily be shown [14] that \tilde{x} is given by

$$\tilde{x} = x_* + V_m y_m$$

where y_m minimizes

$$\|\beta e_1 - \bar{H}_m y_m\| \quad \text{with } \beta \equiv \|r_*\|.$$

This is the generalized minimum residual method (GMRES) algorithm proposed by Saad and Schultz [14]. The new hybrid method will therefore have the following general structure.

THE HYBRID ALGORITHM.

1. *Start:* Choose x_0 , compute $r_0 := f - Ax_0$.

Until convergence do:

2. *Adaptive step:* Set $v_1 := r_0 / \|r_0\|$. Perform k steps of the Arnoldi/GMRES process i.e. compute the GMRES improved solution \tilde{x} from x_0 , and the eigenvalue estimates from the Hessenberg matrix H_m . From the computed eigenvalues get a set H containing the spectrum of A but not the origin. Compute the least squares polynomial s_n of degree $n-1$ based on H . Set $x_0 := \tilde{x}$, $r_0 := f - Ax_0$.

3. *Polynomial iteration:* Compute $x_n := x_0 + s_n(A)r_0$. If satisfied stop else set $x_0 := x_n$, compute $r_0 := f - Ax_0$ and go to 2.

We would like to indicate how the vector x_n is computed in step 3 of the algorithm. Recall that s_n is computed in the form of the expansion (20) in the basis of Chebyshev polynomials t_i which satisfy the three term recurrence relation (18). We therefore need

to compute the vectors $w_i = t_i(A)r_0$ and simultaneously accumulate the linear combination (20) as follows:

1. *Start:* $w_1 := r_0$, $x_0 := 0$.
2. *Iterate:* for $i = 1, 2, \dots, n-1$ do:

$$(25) \quad w_{i+1} := \frac{1}{\beta_{i+1}} [Aw_i - \alpha_i w_i - \delta_i w_{i-1}],$$

$$(26) \quad x_{i+1} := x_i + \eta_{i+1} w_{i+1}.$$

The above iteration requires four vectors of storage (one for x , one for forming the products Av , and two for saving w_i and w_{i-1}). Each step requires $4N$ multiplications. Note that a more economical version that will cost $3N$ multiplications instead of $4N$ consists in computing directly the sequence $\eta_{i+1}w_{i+1}$ instead of w_i , in order to save N multiplications in (26).

Clearly, one is limited in choosing the degree of the polynomial iteration by the fact that the moment matrix becomes difficult to compute and to factor as the degree n increases. A classical solution to this problem is to compound several times a small degree polynomial. This is done by performing the iteration (2) several times, restarting with $x_0 := x_n$, $r_0 := r_n$ after each inner loop.

4. Numerical experiments. The tests reported in this section have been performed on a VAX-11/780 using double precision corresponding to a unit round-off of nearly 6.93×10^{-18} . We compare the performances of the least squares hybrid method described in § 3 with competitive methods, including the nonsymmetric conjugate gradient-like methods, Manteuffel's algorithm [6] and the hybrid Chebyshev-GMRES method [3].

In order to be able to build matrices with specified shapes of spectra, we begin with a class of test matrices which are block diagonals with 2×2 or 1×1 diagonal blocks. To prevent the matrices from being normal each block is of the form

$$\begin{bmatrix} a & b/2 \\ -2b & a \end{bmatrix}$$

and has eigenvalues $a \pm ib$. The eigenvalues are chosen to randomly fill the unions of rectangles. Thus, the first test matrix is 200 by 200 and is selected to have 100 eigenvalues in the rectangle R_1 with vertices at $0.3 \pm 5i$, $0.5 \pm 5i$, and 100 eigenvalues in the rectangle R_2 with vertices $0.5 \pm 0.1i$, $5.0 \pm 0.1i$.

In order to make a fair comparison of several methods, we simulate a five point matrix by counting each matrix by vector product as costing $5N$ multiplications. Indeed, it would be unfair to compare the total number of multiplications in this example because the matrix by vector multiplication costs very little ($2N$ at most). We compare the following methods:

1. The hybrid least squares method (CHEBLS), using 10 Arnoldi vectors. The polynomial s_n is of degree $n = 60$, and is obtained by compounding four times the least squares polynomial of degree 15.
2. The hybrid least squares method using the “exact” region H consisting of the two rectangles R_1 and R_2 . The polynomial is again of degree 60 and is obtained by compounding polynomials of degree 15.
3. The hybrid Chebyshev-GMRES method (CHEBA), using 10 Arnoldi vectors and a maximum number of steps before adapting of 60.
4. The restarted GMRES method [14] using 10 vectors, and then five vectors.

The iteration is stopped as soon as the residual norm is decreased by a factor of 10^{-5} . The right-hand side of the system is random and the initial vector is taken to be

the null vector. In both methods 1 and 3 the eigenvalues are computed dynamically and the convex hull is constructed from these eigenvalue estimates. In method 2, the optimal region $H = R_1 UR_2$ is provided and the least squares polynomial is computed from that region.

We have also tested Manteuffel's adaptive Chebyshev method (CHEB thereafter) but without success as some of the computed eigenvalues have negative real parts thus causing the method to break down. Note that the matrix of this example is not positive real which explains why Manteuffel's adaptive code produced eigenvalues with negative parts. This could also happen with any of the adaptive methods including CHEBA and CHEBLS. If such a situation arises CHEB and CHEBA would fail but CHEBLS would compute two convex hull regions, one on each side of the imaginary axis, and would continue.

The plot in Fig. 4.1 shows the performances of the above four methods. The final upper half part of the convex hull produced by the adaptive Chebyshev least squares method is a triangle having vertices at 0.24 , $0.24 + 4.9i$, and 4.94 . Despite the difference in the regions used in methods 1 and 2, the convergence behavior is not too different.

In the second examples the matrix A is of dimension 100 and is defined in a way similar to the previous one, with 20 eigenvalues enclosed in the rectangle R_1 with vertices $-1.0 \pm 0.1i$, $-030 \pm 0.1i$, and 80 eigenvalues in the rectangle R_2 , having vertices $0.1 \pm 0.1i$, $4.0 \pm 0.1i$. Since the matrix has eigenvalues with both negative and positive real parts, we cannot use the other adaptive Chebyshev methods CHEB and CHEBA. We have compared the hybrid Chebyshev least squares method using 10 vectors versus GMRES(10), ORTHOMIN(5), ORTHOMIN(10). The least squares polynomial is

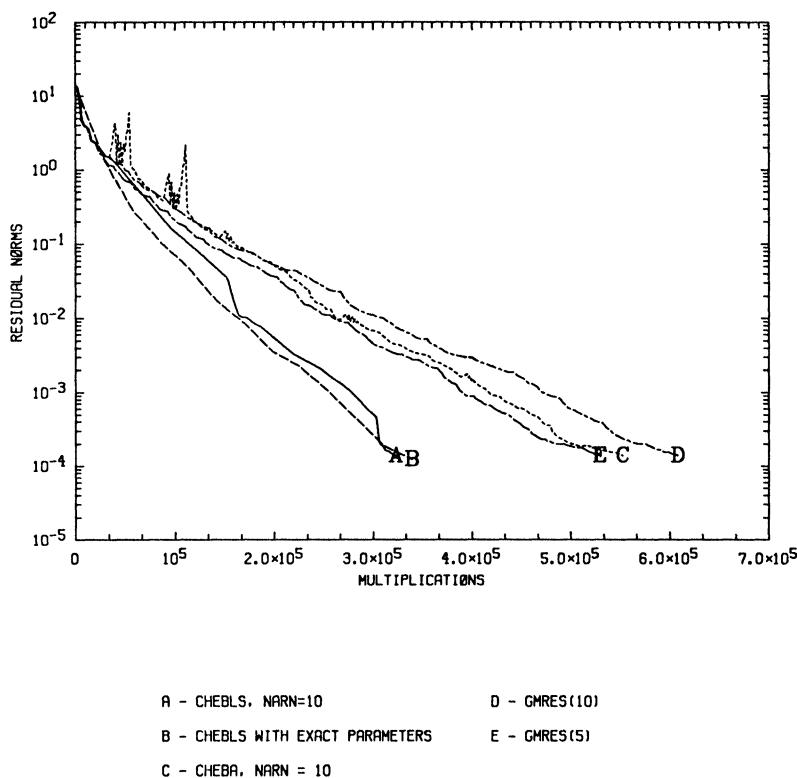


FIG. 4.1. Chebyshev least squares method versus hybrid Chebyshev and GMRES.

again of degree 60 and is obtained by compounding four times a least squares polynomial of degree 15. Note that GMRES(10), ORTHOMIN(5) and CHEBLS all use about the same storage while ORTHOMIN(10) uses about twice as much. The results are plotted in Fig. 4.2. Observe the nonconverging behavior displayed by ORTHOMIN(5) and ORTHOMIN(10) which is quite common for nonpositive real matrices. Although this may also happen with GMRES, our experience is that it is more common with ORTHOMIN.

In this example the conjugate gradient method applied to the normal equations is faster than any of the other methods. Note however that as soon as good eigenvalue estimates are found, CHEBLS outpaces the conjugate gradient method.

The third example is a more realistic one and is derived from the five point discretization of the following partial differential equation which was described in Elman's thesis [2]

$$(27) \quad -(bu_x)_x - (cu_x)_x + du_x + (du)_x + eu_y + (eu)_y + fu = g$$

on the unit square, where

$$\begin{aligned} b(x, y) &= e^{-xy}, & c(x, y) &= e^{xy}, & d(x, y) &= \beta(x + y), \\ e(x, y) &= \gamma(x + y) & \text{and} & f(x, y) &= 1/(1 + x + y), \end{aligned}$$

subject to Dirichlet boundary conditions $u = 0$ on the boundary. The right-hand side g was chosen so that the solution is known to be $x e^{xy} \sin(\pi x) \sin(\pi y)$.

We take 40 interior nodes on each side of the square and $\gamma = 30$, $\beta = -10$. This yields a matrix of dimension $N = 1600$. The system is preconditioned by the MILU

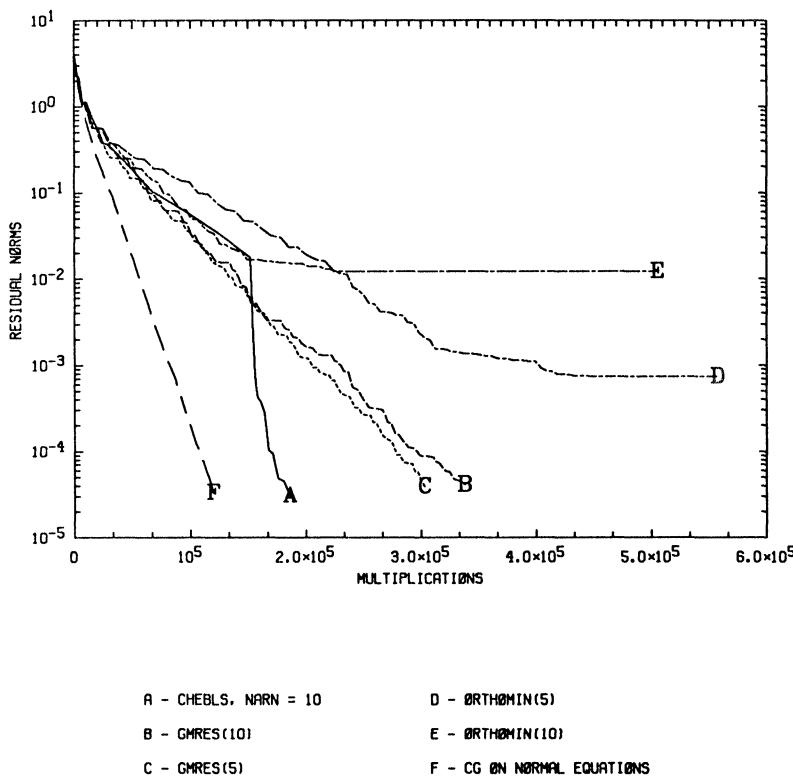


FIG. 4.2. Chebyshev least squares method versus GMRES and ORTHOMIN.

preconditioning applied on the right, i.e., we solve $AM^{-1}(Mx) = f$ where M is some approximation to A^{-1} provided by an approximate LU factorization of A , see [2]. The process is stopped as soon as the residual norm is reduced by a factor of $\varepsilon = 10^{-6}$. The plot in Fig. 4.3 compares the results obtained for CHEBLS, CHEBA, CHEB, all using 7 Arnoldi vectors, GMRES(10), ORTHOMIN(10) and the conjugate gradient method applied to the normal equations. The polynomial based methods perform better than the conjugate gradient type methods. An interesting observation concerning this problem is that convergence is faster with the ILU preconditioning than with the more elaborate MILU. This fact is illustrated in the plot of Fig. 4.3 where we added the results for ORTHOMIN(1) with the ILU preconditioning (curve E).

Generally, one observes that when a problem is well behaved then the methods based on approximations from the same Krylov subspace have similar performances. The MILU-preconditioned problem of this example is well conditioned in the sense that the ratio of its largest to its smallest eigenvalue is not large, as can be judged from the (simplified) convex hull that was computed from CHEBLS:

$$\begin{aligned} h_0 &= 1.138, & h_1 &= 1.138 + 0.644i, & h_2 &= 1.525 + 1.967i, \\ h_3 &= 3.053 + 3.100i, & h_4 &= 10.5448. \end{aligned}$$

Note that the eigenvalues h_i from which the above convex hull is built have at least three digits of accuracy.

The reason why ORTHOMIN and the other conjugate gradient-like methods do not perform as well as might be expected from this nice distribution is simply due to the fact that the preconditioned matrix is not positive real. Note that Arnoldi's method

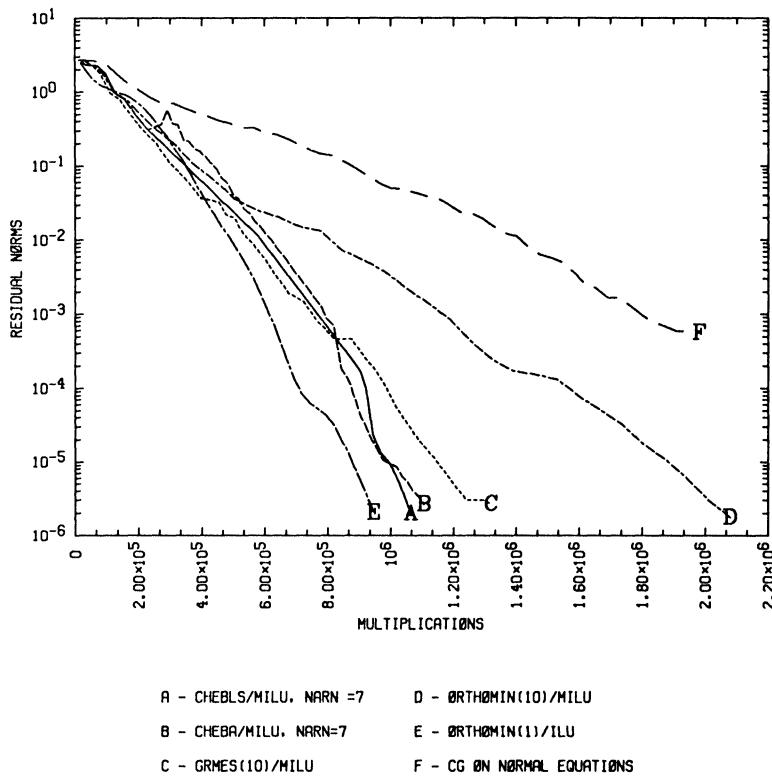


FIG. 4.3. Test problem associated with PDE problem (27).

for computing eigenvalues may then yield eigenvalues with negative real parts, because the computed eigenvalues are only known to lie inside the field of values of the matrix. Experience seems to suggest that this risk can be lessened by taking a larger number of vectors in the projection step, but no proof of this fact seems to be available. Practically, one can be more selective and base the convex hull only on the eigenvalues that have a certain minimum accuracy. Note that the residual norms of the eigenpairs are available from the Hessenberg matrix and so the eigenvectors of the iteration matrix need not be computed [12].

Acknowledgment. The author would like to thank Professor Walter Gautschi for his helpful comments on bibliographical matters.

REFERENCES

- [1] W. E. ARNOLDI, *The principle of minimized iteration in the solution of the matrix eigenvalue problem*, Quart. Appl. Math., 9 (1951), pp. 17–29.
- [2] H. C. ELMAN, *Iterative methods for large sparse nonsymmetric systems of linear equations*, Ph.D. thesis, Yale University, New Haven, Connecticut, Computer Science Dept., 1982.
- [3] H. C. ELMAN, Y. SAAD AND P. SAYLOR, *A hybrid Chebyshev-Krylov subspace algorithm for solving nonsymmetric systems of linear equations*, Technical Report YALU/DCS/TR-301, Yale University, New Haven, Connecticut, 1984.
- [4] W. GAUTSCHI, *On generating orthogonal polynomials*, SIAM J. Sci. Statist. Comput., 3 (1982), pp. 289–317.
- [5] A. L. HAGEMAN AND D. M. YOUNG, *Applied Iterative Methods*, Academic Press, New York, 1981.
- [6] T. A. MANTEUFFEL, *An iterative method for solving nonsymmetric linear systems with dynamic estimation of parameters*, Technical Report UIUCDCS-75-758, University of Illinois at Urbana-Champaign, Ph.D. dissertation, Urbana-Champaign, Illinois, 1975.
- [7] ———, *The Tchebychev iteration for nonsymmetric linear systems*, Numer. Math., 28 (1977), pp. 307–327.
- [8] ———, *Adaptive procedure for estimation of parameter for the nonsymmetric Tchebychev iteration*, Numer. Math., 28 (1978), pp. 187–208.
- [9] Y. SAAD, *Least squares polynomials in the complex plane with applications to solving sparse nonsymmetric matrix problems*, Technical Report 276, Yale University, New Haven, Connecticut, 1983.
- [10] ———, *On the condition numbers of modified moment matrices arising in least squares approximation in the complex plane*, Numer. Math., 48 (1986), pp. 337–347.
- [11] ———, *Chebyshev acceleration techniques for solving nonsymmetric eigenvalue problems*, Math. Comp., 42 (1984), pp. 567–588.
- [12] ———, *Variations on Arnoldi's method for computing eigenelements of large unsymmetric matrices*, Linear Algebra Appl., 34 (1980), pp. 269–295.
- [13] ———, *Iterative solution of indefinite symmetric systems by methods using orthogonal polynomials over two disjoint intervals*, this Journal, 20 (1983), pp. 784–811.
- [14] Y. SAAD AND M. H. SCHULTZ, *GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems*, Technical Report 254, Yale University, New Haven, Connecticut, 1983; to appear in SIAM J. Sci. Statist. Comput.
- [15] R. A. SACK AND A. F. DONOVAN, *An algorithm for Gaussian quadrature given modified moments*, Numer. Math., 18 (1971), pp. 465–478.
- [16] D. C. SMOLARSKI, *Optimum semi-iterative methods for the solution of any linear algebraic system with a square matrix*, Technical Report UIUCDCS-R-81-1077, University of Illinois at Urbana-Champaign, Ph.D. thesis, Urbana-Champaign, Illinois, 1981.
- [17] D. C. SMOLARSKI AND P. E. SAYLOR, *Optimum parameters for the solution of linear equations by Richardson's iteration*, 1982, unpublished paper.
- [18] E. L. STIEFEL, *Kernel polynomials in linear algebra and their applications*, U.S. NBS Appl. Math. Series, 49 (1958), pp. 1–24.
- [19] H. E. WRIGLEY, *Accelerating the Jacobi method for solving simultaneous equations by Chebyshev extrapolation when the eigenvalues of the iteration matrix are complex*, Comput. J., 6 (1963), pp. 169–176.