Chapter 4

Effects of Finite Precision Arithmetic

In the previous chapter, error bounds were derived for the CG, MINRES, and GMRES algorithms, using the fact that these methods find the *optimal* approximation from a Krylov subspace. In the Arnoldi algorithm, on which the GMRES method is based, all of the Krylov space basis vectors are retained, and a new vector is formed by explicitly orthogonalizing against all previous vectors using the modified Gram-Schmidt procedure. The modified Gram-Schmidt procedure is known to yield nearly orthogonal vectors if the vectors being orthogonalized are not too nearly linearly dependent. In the special case where the vectors are almost linearly dependent, the modified Gram-Schmidt procedure can be replaced by Householder transformations, at the cost of some extra arithmetic [139]. In this case, one would expect the basis vectors generated in the GMRES method to be almost orthogonal and the approximate solution obtained to be nearly optimal, at least in the space spanned by these vectors. For discussions of the effect of rounding errors on the GMRES method, see [33, 70, 2].

This is not the case for the CG and MINRES algorithms, which use short recurrences to generate orthogonal basis vectors for the Krylov subspace. The proof of orthogonality, and hence of the optimality of the approximate solution, relies on induction (e.g., Theorems 2.3.1 and 2.3.2 and the arguments after the Lanczos algorithm in section 2.5), and such arguments may be destroyed by the effects of finite precision arithmetic. In fact, the basis vectors generated by the Lanczos algorithm (or the residual vectors generated by the CG algorithm) in finite precision arithmetic frequently lose orthogonality completely and may even become linearly dependent! In such cases, the approximate solutions generated by the CG and MINRES algorithms are *not* the optimal approximations from the Krylov subspace, and it is not clear that any of the results from Chapter 3 should hold.

In this chapter we show why the nonorthogonal vectors generated by the Lanczos algorithm can still be used effectively for solving linear systems and which of the results from Chapter 3 can and cannot be expected to hold (to a close approximation) in finite precision arithmetic. It is shown that for both the MINRES and CG algorithms, the 2-norm of the residual is essentially determined by the *tridiagonal matrix* produced in the finite precision Lanczos computation. This tridiagonal matrix is, of course, quite different from the one that would be produced in exact arithmetic. It follows, however, that if the same tridiagonal matrix would be produced by the exact Lanczos algorithm applied to some other problem, then exact arithmetic bounds on the residual for that problem will hold for the finite precision computation. In order to establish exact arithmetic bounds for the different problem, it is necessary to have some information about the eigenvalues of the new coefficient matrix. Here we make use of results already established in the literature about the eigenvalues of the new coefficient matrix, but we do not include the proofs.

The analysis presented here is by no means a complete rounding-error analysis of the algorithms given in Chapter 2. As anyone who has done a rounding-error analysis knows, the arguments can quickly become complicated and tedious. Here we attempt to present some of the more interesting aspects of the error analysis for the CG and MINRES algorithms, without becoming bogged down in the details. We consider a hypothetical implementation of these algorithms for which the analysis is easier and refer to the literature for arguments about the precise nature of the roundoff terms at each step.

This analysis deals with the rate at which the A-norm of the error in the CG algorithm and the 2-norm of the residual in the MINRES algorithm are reduced before the ultimately attainable accuracy is achieved. A separate issue is the level of accuracy that can be attained if the iteration is carried out for sufficiently many steps. This question is discussed in section 7.3.

4.1. Some Numerical Examples.

To illustrate the numerical behavior of the CG algorithm of section 2.3, we have applied this algorithm to linear systems $Ax = b$ with coefficient matrices of the form $A = U\Lambda U^H$, where U is a random orthogonal matrix and $\Lambda = \text{diag}(\lambda_1, \ldots, \lambda_n)$, where

$$(4.1) \qquad \lambda_i = \lambda_1 + \frac{i-1}{n-1}(\lambda_n - \lambda_1)\rho^{n-i}, \quad i = 2, \ldots, n-1,$$

and the parameter ρ is chosen between 0 and 1. For $\rho = 1$, the eigenvalues are uniformly spaced, and for smaller values of ρ the eigenvalues are tightly clustered at the lower end of the spectrum and are far apart at the upper end. We set $n = 24$, $\lambda_1 = .001$, and $\lambda_n = 1$. A random right-hand side and zero initial guess were used in all cases. Figure 4.1a shows a plot of the A-norm of the error versus the iteration number for $\rho = .4, .6, .8, 1$. Experiments were performed using double precision Institute of Electrical and Electronics Engineers (IEEE) arithmetic, with machine precision $\epsilon \approx 1.1e-16$. Figure 4.1b shows what these curves would look like if exact arithmetic had been used. (Exact arithmetic can be simulated in the CG algorithm by saving all of the basis vectors in the Lanczos algorithm and explicitly orthogonalizing against them at every step. This is how the data for Figure 4.1b was produced.)

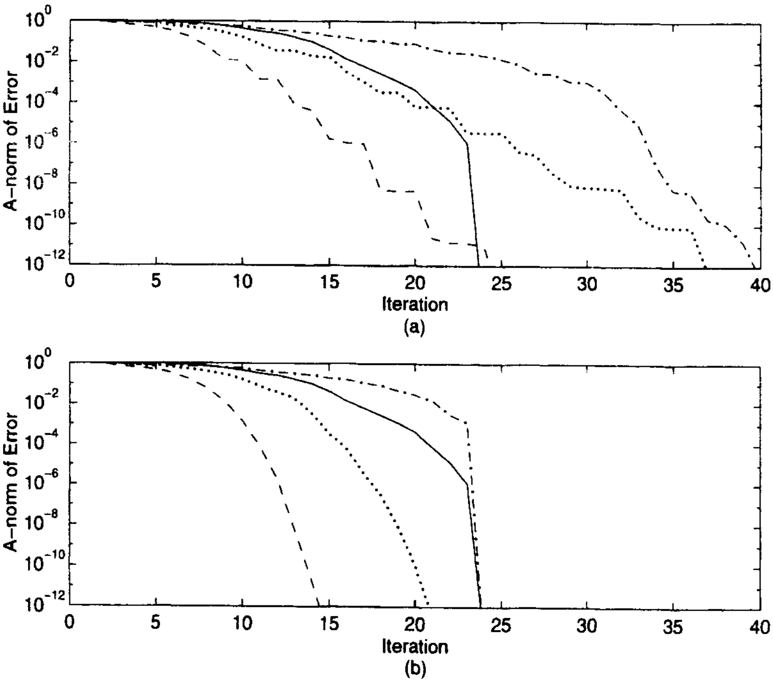

FIG. 4.1. *CG convergence curves for* (a) *finite precision arithmetic and* (b) *exact arithmetic.* $\rho = 1$ *solid,* $\rho = .8$ *dash-dot,* $\rho = .6$ *dotted,* $\rho = .4$ *dashed.*

Note that although the theory of Chapter 3 guarantees that the exact solution is obtained after $n = 24$ steps, the computations with $\rho = .6$ and $\rho = .8$ do not generate good approximate solutions by step 24. It is at about step 31 that the error in the $\rho = .8$ computation begins to decrease rapidly. The $\rho = .4$ computation has reduced the A-norm of the error to $1.e-12$ by step 24, but the corresponding exact arithmetic calculation would have reduced it to this level after just 14 steps. In contrast, for $\rho = 1$ (the case with equally spaced eigenvalues), the exact and finite precision computations behave very similarly. In all cases, the finite precision computation eventually finds a good approximate solution, but it is clear that estimates of the number of iterations required to do so cannot be based on the error bounds of Chapter 3. In this chapter we develop error bounds that hold in finite precision arithmetic.

4.2. The Lanczos Algorithm.

When the Lanczos algorithm is implemented in finite precision arithmetic, the recurrence of section 2.5 is perturbed slightly. It is replaced by a recurrence that can be written in matrix form as

$$AQ_k = Q_k T_k + \beta_k q_{k+1} \xi_k^T + F_k = Q_{k+1} T_{k+1,k} + F_k, \tag{4.2}$$

where the columns of F_k represent the rounding errors at each step. Let ϵ denote the machine precision and define

$$\epsilon_0 \equiv 2(n+4)\epsilon, \quad \epsilon_1 \equiv 2(7 + m \; \| \; |A| \; \| / \|A\|) \; \epsilon, \tag{4.3}$$

where m is the maximum number of nonzeros in any row of A. Under the assumptions that

$$\epsilon_0 < \frac{1}{12}, \quad k(3\epsilon_0 + \epsilon_1) < 1, \tag{4.4}$$

and ignoring higher order terms in ϵ, Paige [109] showed that the rounding error matrix F_k satisfies

$$\|F_k\| \leq \sqrt{k}\, \epsilon_1 \; \|A\|. \tag{4.5}$$

Paige also showed that the coefficient formulas in the Lanczos algorithm can be implemented sufficiently accurately to ensure that

$$|q_j^T q_j - 1| \leq 2\epsilon_0, \tag{4.6}$$

$$\beta_j \leq \|A\| \; (1 + (2n+6)\epsilon + j(3\epsilon_0 + \epsilon_1)). \tag{4.7}$$

We will assume throughout that the inequalities (4.4) and hence (4.5–4.7) hold.

Although the individual roundoff terms are tiny, their effect on the recurrence (4.2) may be great. The Lanczos vectors may lose orthogonality and even become linearly dependent. The recurrence coefficients generated in finite precision arithmetic may be quite different from those that would be generated in exact arithmetic.

4.3. A Hypothetical MINRES/CG Implementation.

Although the computed Lanczos vectors may not be orthogonal, one might still consider using them in the CG or MINRES algorithms for solving linear systems; that is, one could still choose an approximate solution x_k of the form

$$x_k = x_0 + Q_k y_k, \tag{4.8}$$

where y_k solves the least squares problem

$$\min_y \|\beta\xi_1 - T_{k+1,k} y\|, \quad \beta \equiv \|r_0\| \tag{4.9}$$

for the MINRES method or the linear system

$$T_k y = \beta\xi_1 \tag{4.10}$$

for the CG algorithm. Of course, in practice, one does not first compute the Lanczos vectors and then apply formulas (4.8–4.10), since this would require saving all of the Lanczos vectors. Still, it is reasonable to try and separate the effects of roundoff on the three-term Lanczos recurrence from that on other aspects of the (implicit) evaluation of (4.8–4.10). It is the effect of using the nonorthogonal vectors produced by a finite precision Lanczos computation that is analyzed here, so from here on we assume that formulas (4.8–4.10) hold exactly, where Q_k, T_k, and $T_{k+1,k}$ satisfy (4.2).

The residual in the CG algorithm, which we denote here as r_k^C, then satisfies

$$\begin{aligned} r_k^C &= r_0 - AQ_k y_k^C = r_0 - (Q_k T_k + \beta_k q_{k+1}\xi_k^T + F_k) y_k^C \\ &= -\beta_k q_{k+1} \xi_k^T y_k^C - F_k y_k^C \\ &= -\beta\, (\beta_k q_{k+1} \xi_k^T T_k^{-1} \xi_1 + F_k y_k^C / \beta), \end{aligned}$$

where y_k^C denotes the solution to (4.10). The 2-norm of the residual satisfies

$$\begin{aligned} &\|q_{k+1}\|\ |\beta_k \xi_k^T T_k^{-1} \xi_1| - \|F_k\|\ \|y_k^C\|/\beta \le \|r_k^C\|/\|r_0\| \\ (4.11)\qquad &\le \|q_{k+1}\|\ |\beta_k \xi_k^T T_k^{-1} \xi_1| + \|F_k\|\ \|y_k^C\|/\beta. \end{aligned}$$

Using (4.5) and (4.6), this becomes

$$\begin{aligned} &\sqrt{1+2\epsilon_0}\ |\beta_k \xi_k^T T_k^{-1} \xi_1| - \sqrt{k}\ \epsilon_1\ \|A\|\ \|y_k^C\|/\beta \le \|r_k^C\|/\|r_0\| \\ (4.12)\qquad &\le \sqrt{1+2\epsilon_0}\ |\beta_k \xi_k^T T_k^{-1} \xi_1| + \sqrt{k}\ \epsilon_1\ \|A\|\ \|y_k^C\|/\beta. \end{aligned}$$

It follows that at steps k, where $\sqrt{k}\ \epsilon_1\ \|A\|\ \|y_k^C\|/\beta$ is much smaller than the residual norm, the 2-norm of the residual is essentially determined by the tridiagonal matrix T_k and the next recurrence coefficient β_k.

The residual in the MINRES algorithm, which we denote here as r_k^M, satisfies

$$\begin{aligned} r_k^M &= r_0 - AQ_k y_k^M = r_0 - (Q_{k+1} T_{k+1,k} + F_k) y_k^M \\ &= Q_{k+1}(\beta \xi_1 - T_{k+1,k} y_k^M) - F_k y_k^M, \end{aligned}$$

where y_k^M denotes the solution to (4.9). The 2-norm of the residual satisfies

$$(4.13)\qquad \|r_k^M\|/\|r_0\| \le \|Q_{k+1}\|\ \|\xi_1 - T_{k+1,k} y_k^M/\beta\| + \|F_k\|\ \|y_k^M\|/\beta.$$

It follows from (4.6) that

$$\|Q_{k+1}\| \le \sqrt{(1+2\epsilon_0)(k+1)},$$

so, with (4.5), we have

$$\begin{aligned} &\|r_k^M\|/\|r_0\| \le \sqrt{(1+2\epsilon_0)(k+1)}\ \|\xi_1 - T_{k+1,k} y_k^M/\beta\| \\ (4.14)\qquad &+\sqrt{k}\ \epsilon_1\ \|A\|\ \|y_k^M\|/\beta. \end{aligned}$$

It follows that at steps k, where $\sqrt{k}\ \epsilon_1\ \|A\|\ \|y_k^M\|/\beta$ is tiny compared to the residual norm, the 2-norm of the residual is essentially bounded, to within a possible factor of $\sqrt{k+1}$ (which is usually an overestimate), by an expression involving only the $k+1$-by-k tridiagonal matrix $T_{k+1,k}$.

Thus, for both the MINRES and CG algorithms, the 2-norm of the residual (or at least a realistic bound on the 2-norm of the residual) is essentially determined by the recurrence coefficients computed in the finite precision

Lanczos computation and stored in the tridiagonal matrix $T_{k+1,k}$. Suppose the exact Lanczos algorithm, applied to a matrix (or linear operator) $\mathcal{A}$ with initial vector φ_1, generates the same tridiagonal matrix $T_{k+1,k}$. It would follow that the 2-norm of the residual r_k^M or r_k^C in the finite precision computation would be approximately the same as the 2-norm of the residual ν_k^M or ν_k^C in the exact MINRES or CG algorithm for solving the linear system (or operator equation) $\mathcal{A}\chi = \varphi$, with right-hand side $\varphi = \beta\varphi_1$; in this case, we would have

$$\|\nu_k^C\|/\|r_0\| = |\beta_k \xi_k^T T_k^{-1} \xi_1|, \qquad \|\nu_k^M\|/\|r_0\| = \|\xi_1 - T_{k+1,k} y_k^M/\beta\|.$$

Compare with (4.12) and (4.14).

Note also that if T is any Hermitian tridiagonal matrix (even an infinite one) whose upper left $k+1$-by-k block is $T_{k+1,k}$, then the exact Lanczos algorithm applied to T with initial vector ξ_1 will generate the matrix $T_{k+1,k}$ at step k. This follows because the reduction of a Hermitian matrix to tridiagonal form (with nonnegative off-diagonal entries) is uniquely determined once the initial vector is set.

With this observation, we can now use results about the convergence of the exact MINRES and CG algorithms applied to any such matrix T to derive bounds on the residuals r_k^M and r_k^C in the finite precision computation. To do this, one must have some information about the eigenvalues of such a matrix T.

4.4. A Matrix Completion Problem.

With the arguments of the previous section, the problem of bounding the residual norm in finite precision CG and MINRES computations becomes a matrix completion problem: given the $k+1$-by-k tridiagonal matrix $T_{k+1,k}$ generated by a finite precision Lanczos computation, find a Hermitian tridiagonal matrix T with $T_{k+1,k}$ as its upper left block,

$$T = \begin{pmatrix}
\alpha_1 & \beta_1 & & & & & & \\
\beta_1 & \ddots & \ddots & & & & & \\
 & \ddots & \ddots & \beta_{k-1} & & & & \\
 & & \beta_{k-1} & \alpha_k & \beta_k & & & \\
 & & & \beta_k & * & * & & \\
 & & & & * & \ddots & \ddots & \\
 & & & & & \ddots & \ddots & * \\
 & & & & & & * & *
\end{pmatrix},$$

whose eigenvalues are related to those of A in such a way that the exact arithmetic error bounds of Chapter 3 yield useful results about the convergence of the exact CG or MINRES algorithms applied to linear systems with coefficient matrix T. In this section we state such results but refer the reader to the literature for their proofs.

4.4.1. Paige's Theorem.

The following result of Paige [110] shows that the eigenvalues of T_{k+1}, the tridiagonal matrix generated at step $k+1$ of a finite precision Lanczos computation, lie essentially between the largest and smallest eigenvalues of A.

THEOREM 4.4.1 (Paige). *The eigenvalues $\theta_i^{(j)}$, $i = 1, \ldots, j$ of the tridiagonal matrix T_j satisfy*

$$\lambda_1 - j^{5/2}\epsilon_2\|A\| \le \theta_i^{(j)} \le \lambda_n + j^{5/2}\epsilon_2\|A\|, \quad \epsilon_2 = \sqrt{2}\ \max\{6\epsilon_0, \epsilon_1\}, \tag{4.15}$$

where λ_1 is the smallest eigenvalue of A, λ_n is the largest eigenvalue of A, and ϵ_0 and ϵ_1 are defined in (4.3).

Using this result with the arguments of the previous section, we obtain the following result for Hermitian positive definite matrices A.

THEOREM 4.4.2. *Let A be a Hermitian positive definite matrix with eigenvalues $\lambda_1 \le \cdots \le \lambda_n$ and assume that $\lambda_1 - (k+1)^{5/2}\epsilon_2\|A\| > 0$. Let r_k^M and r_k^C denote the residuals at step k of the MINRES and CG computations satisfying* (4.8) *and* (4.9) *or* (4.10), *respectively, where Q_k, T_k, and $T_{k+1,k}$ satisfy* (4.2). *Then*

$$\|r_k^C\|/\|r_0\| \le \sqrt{1+2\epsilon_0}\ 2\sqrt{\tilde{\kappa}}\left(\frac{\sqrt{\tilde{\kappa}}-1}{\sqrt{\tilde{\kappa}}+1}\right)^k + \sqrt{k}\ \tilde{\kappa}\ \epsilon_1, \tag{4.16}$$

$$\|r_k^M\|/\|r_0\| \le \sqrt{(1+2\epsilon_0)(k+1)}\ 2\left(\frac{\sqrt{\tilde{\kappa}}-1}{\sqrt{\tilde{\kappa}}+1}\right)^k + \sqrt{k}\ \tilde{\kappa}\ \epsilon_1, \tag{4.17}$$

where

$$\tilde{\kappa} = \frac{\lambda_n + (k+1)^{5/2}\epsilon_2\|A\|}{\lambda_1 - (k+1)^{5/2}\epsilon_2\|A\|} \tag{4.18}$$

and ϵ_0, ϵ_1, and ϵ_2 are defined in (4.3) *and* (4.15).

Proof. It follows from Theorem 4.4.1 that T_k is nonsingular, and since $y_k^C = T_k^{-1}\beta\xi_1$, we have for the second term on the right-hand side of (4.12)

$$\|A\|\ \|y_k^C\|/\beta \le \|A\|\ \|T_k^{-1}\| \le \frac{\lambda_n}{\lambda_1 - k^{5/2}\epsilon_2\|A\|} \le \tilde{\kappa}.$$

Since the expression $|\beta_k\xi_k^T T_k^{-1}\xi_1|$ in (4.12) is the size of the residual at step k of the exact CG algorithm applied to a linear system with coefficient matrix T_{k+1} and right-hand side ξ_1 and since the eigenvalues of T_{k+1} satisfy (4.15), it follows from (3.8) that

$$|\beta_k\xi_k^T T_k^{-1}\xi_1| \le 2\sqrt{\tilde{\kappa}}\left(\frac{\sqrt{\tilde{\kappa}}-1}{\sqrt{\tilde{\kappa}}+1}\right)^k,$$

where $\tilde{\kappa}$ is given by (4.18). Here we have used the fact that since the expression in (3.8) bounds the reduction in the T_{k+1}-norm of the error in the exact

CG iterate, the reduction in the 2-norm of the residual for this exact CG iterate is bounded by $\sqrt{\tilde{\kappa}}$ times the expression in (3.8); i.e., $\|Bv\|/\|Bw\| \leq \sqrt{\kappa(B)}\ \|v\|_B/\|w\|_B$ for any vectors v and w and any positive definite matrix B. Making these substitutions into (4.12) gives the desired result (4.16).

For the MINRES algorithm, it can be seen from the Cauchy interlace theorem (Theorem 1.3.12) applied to $T_{k+1,k}T_{k+1,k}^H$ that the smallest singular value of $T_{k+1,k}$ is greater than or equal to the smallest eigenvalue of T_k. Consequently, we have

$$\|y_k^M\| \leq \beta\ \|T_k^{-1}\|,$$

so, similar to the CG algorithm, the second term on the right-hand side of (4.14) satisfies

$$\|A\|\ \|y_k^M\|/\beta \leq \|A\|\ \|T_k^{-1}\| \leq \frac{\lambda_n}{\lambda_1 - k^{5/2}\epsilon_2\|A\|} \leq \tilde{\kappa}.$$

Since the expression $\|\xi_1 - T_{k+1,k}y_k^M/\beta\|$ in (4.14) is the size of the residual at step k of the exact MINRES algorithm applied to the linear system $T_{k+1}\chi = \xi_1$, where the eigenvalues of T_{k+1} satisfy (4.15), it follows from (3.12) that

$$\|\xi_1 - T_{k+1,k}y_k^M/\beta\| \leq 2\left(\frac{\sqrt{\tilde{\kappa}}-1}{\sqrt{\tilde{\kappa}}+1}\right)^k.$$

Making these substitutions into (4.14) gives the desired result (4.17). □

Theorem 4.4.2 shows that, at least to a close approximation, the exact arithmetic residual bounds based on the size of the Chebyshev polynomial on the *interval* from the smallest to the largest eigenvalue of A hold in finite precision arithmetic as well. Exact arithmetic bounds such as (3.11) and (3.12) for $\ell > 0$, based on approximation on *discrete* subsets of the eigenvalues of A may fail, however, as may the sharp bounds (3.6) and (3.7). This was illustrated in section 4.1. Still, stronger bounds than (4.16) and (4.17) may hold in finite precision arithmetic, and such bounds are derived in the next subsection.

4.4.2. A Different Matrix Completion.

Paige's theorem about the eigenvalues of T_{k+1} lying essentially between the smallest and largest eigenvalues of A is of little use in the case of indefinite A, since in that case, T_{k+1} could be singular. Moreover, we would like to find a completion T of $T_{k+1,k}$ whose eigenvalues can be more closely related to the discrete eigenvalues of A in order to obtain finite precision analogues of the sharp error bounds (3.6) and (3.7).

It was shown by Greenbaum that $T_{k+1,k}$ can be extended to a larger Hermitian tridiagonal matrix T whose eigenvalues all lie in tiny intervals about the eigenvalues of A [65], the size of the intervals being a function of the machine precision. Unfortunately, the proven bound on the interval size appears to be a large overestimate. The bound on the interval size established in [65] involves a number of constants as well as a factor of the form $n^3k^2\sqrt{\epsilon}\|A\|$ or, in some cases, $n^3k\epsilon^{1/4}\|A\|$, but better bounds are believed possible.

Suppose the eigenvalues of such a matrix T have been shown to lie in intervals of width δ about the eigenvalues of A. One can then relate the size of the residual at step k of a finite precision computation to the maximum value of the minimax polynomial on the union of tiny intervals containing the eigenvalues of T, using the same types of arguments as given in Theorem 4.4.2.

THEOREM 4.4.3. *Let A be a Hermitian matrix with eigenvalues $\lambda_1 \leq \cdots \leq \lambda_n$ and let $T_{k+1,k}$ be the $k+1$-by-k tridiagonal matrix generated by a finite precision Lanczos computation. Assume that there exists a Hermitian tridiagonal matrix T, with $T_{k+1,k}$ as its upper left $k+1$-by-k block, whose eigenvalues all lie in the intervals*

$$S = \bigcup_{i=1}^{n} [\lambda_i - \delta, \lambda_i + \delta], \tag{4.19}$$

where none of the intervals contains the origin. Let d denote the distance from the origin to the set S. Then the MINRES residual r_k^M satisfies

$$\begin{aligned} \|r_k^M\|/\|r_0\| \leq \sqrt{(1+2\epsilon_0)(k+1)}\ \min_{p_k}\ \max_{z\in S}|p_k(z)| \\ +2\sqrt{k}\ (\lambda_n/d)\ \epsilon_1. \end{aligned} \tag{4.20}$$

If A is positive definite, then the CG residual r_k^C satisfies

$$\begin{aligned} \|r_k^C\|/\|r_0\| \leq \sqrt{1+2\epsilon_0}\ \sqrt{(\lambda_n+\delta)/d}\ \min_{p_k}\ \max_{z\in S}|p_k(z)| \\ +\sqrt{k}\ (\lambda_n/d)\ \epsilon_1. \end{aligned} \tag{4.21}$$

Proof. Since the expression $\|\xi_1 - T_{k+1,k}y_k^M/\beta\|$ in (4.14) is the size of the residual at step k of the exact MINRES algorithm applied to the linear system $T\chi = \xi_1$, where the eigenvalues of T lie in S, it follows from (3.7) that

$$\|\xi_1 - T_{k+1,k}y_k^M/\beta\| \leq \min_{p_k}\ \max_{z\in S}|p_k(z)|.$$

To bound the second term in (4.14), note that the approximate solution generated at step k of this corresponding exact MINRES calculation is of the form $\chi_k = \mathcal{Q}_k y_k^M/\beta$, where the columns of $\mathcal{Q}_k$ are orthonormal and the vector y_k^M is the same one generated by the finite precision computation. It follows that $\|y_k^M\|/\beta = \|\chi_k\|$. Since the 2-norm of the residual decreases monotonically in the exact algorithm, we have

$$\|\xi_1 - T\chi_k\| \leq 1 \ \Rightarrow\ \|T^{-1}\xi_1 - \chi_k\| \leq \|T^{-1}\| \ \Rightarrow\ \|\chi_k\| \leq 2\|T^{-1}\|.$$

Making these substitutions in (4.14) gives

$$\|r_k^M\|/\|r_0\| \leq \sqrt{(1+2\epsilon_0)(k+1)}\ \min_{p_k}\ \max_{z\in S}|p_k(z)| + 2\sqrt{k}\ \epsilon_1\ \|A\|\ \|T^{-1}\|,$$

from which the desired result (4.20) follows.

When A, and hence T, is positive definite, the expression $|\beta_k \xi_k^T T_k^{-1} \xi_1|$ in (4.12) is the size of the residual at step k of the exact CG algorithm applied to the linear system $T\chi = \xi_1$. It follows from (3.6) that

$$|\beta_k \xi_k^T T_k^{-1} \xi_1| \leq \sqrt{\kappa(T)} \min_{p_k} \max_{z \in S} |p_k(z)|,$$

where the factor $\sqrt{\kappa(T)} = \sqrt{(\lambda_n + \delta)/d}$ must be included, since this gives a bound on the 2-norm of the residual instead of the T-norm of the error. The second term in (4.12) can be bounded as in Theorem 4.4.2. Since $y_k^C = T_k^{-1} \beta \xi_1$ and since, by the Cauchy interlace theorem, the smallest eigenvalue of T_k is greater than or equal to that of T, we have

$$\|A\| \, \|y_k^C\| / \beta \leq \|A\| \, \|T_k^{-1}\| \leq \lambda_n / d.$$

Making these substitutions in (4.12) gives the desired result (4.21). □

Theorem 4.4.3 shows that, to a close approximation, the exact arithmetic residual bounds based on the size of the minimax polynomial on the *discrete* set of eigenvalues of A can be replaced, in finite precision arithmetic, by the size of the minimax polynomial on the union of *tiny intervals* in (4.19). Bounds such as (3.11) and (3.12) for $\ell > 0$ will not hold in finite precision arithmetic. Instead, if A has a few large outlying eigenvalues, one must consider a polynomial that is the product of one with enough roots in the outlying intervals to ensure that it is tiny throughout these intervals, with a lower-degree Chebyshev polynomial on the remainder of the spectrum. The maximum value of this polynomial throughout the set S in (4.19) provides an error bound that holds in finite precision arithmetic. It is still advantageous, in finite precision arithmetic, to have most eigenvalues concentrated in a small interval with just a few outliers (as opposed to having eigenvalues everywhere throughout the larger interval), but the advantages are less than in exact arithmetic.

It is shown in [65] that not only the residual norm bound (4.21) but also the corresponding bound on the A-norm of the error,

$$\|e_k^C\|_A / \|e_0\|_A \leq \min_{p_k} \max_{z \in S} |p_k(z)|, \tag{4.22}$$

holds to a close approximation in finite precision arithmetic. In Figure 4.2, this error bound is plotted along with the actual A-norm of the error in a finite precision computation with a random right-hand side and zero initial guess for the case $\rho = .6$ described in section 4.1. The interval width δ was taken to be $1.e-15$, or about 10ϵ. For comparison, the sharp error bound for exact arithmetic (3.6) is also shown in Figure 4.2. It is evident that the bound (4.22) is applicable to the finite precision computation and that it gives a reasonable estimate of the actual error when the initial residual is random.

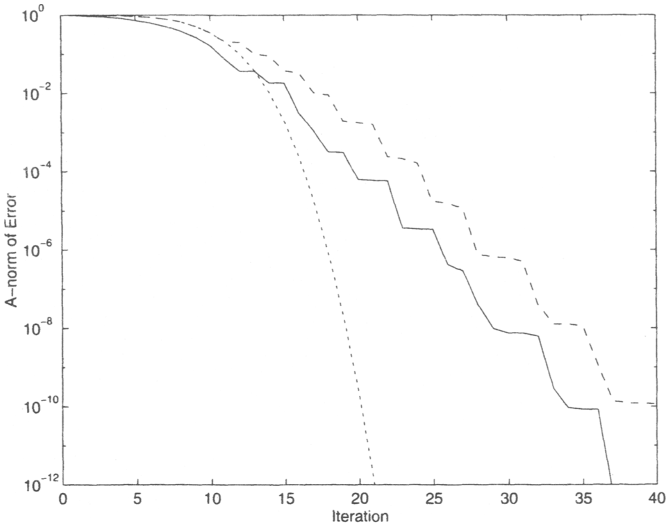

FIG. 4.2. *Exact arithmetic error bound (dotted), finite precision arithmetic error bound (assuming $\delta = 1.e - 15$) (dashed), and actual error in a finite precision computation (solid).*

4.5. Orthogonal Polynomials.

The theorems of the previous section identified the behavior of the first k steps of the CG and MINRES algorithms in finite precision arithmetic with that of the first k steps of the exact algorithms applied to a different problem. (That is, the tridiagonal matrices generated during the first k steps are the same, and so the residual norms agree to within the factors given in the theorems.) Of course, if the bound δ on the interval size in Theorem 4.4.3 were *independent* of k, then this would imply that the identity (between tridiagonal matrices generated in finite precision arithmetic and those generated by the exact algorithm applied to a linear operator with eigenvalues contained in intervals of width δ about the eigenvalues of A) would hold for arbitrarily many steps. It is not known whether the assumption of Theorem 4.4.3 can be satisfied for some small value of δ that does not depend on k.

The analysis of section 4.4 is somewhat unusual in linear algebra. Normally, the approximate solution generated by a finite precision computation is identified with the exact solution of a nearby problem of the *same dimension*. The matrix T in Theorem 4.4.3 can be of any dimension greater than or equal to $k+1$. It represents a *nearby* problem only in the sense that its eigenvalues lie close to those of A. The arguments of the previous sections have a somewhat more natural interpretation in terms of orthogonal polynomials.

The 3-term recurrence of the Lanczos algorithm (in exact arithmetic) implicitly constructs the orthonormal polynomials for a certain set of weights on the eigenvalues of the matrix – the weights being the squared components of the initial vector in the direction of each eigenvector of A. To see this, let $A = U \Lambda U^H$ be an eigendecomposition of A and let $\hat{q}_j = U^H q_j$, where the vectors q_j, $j = 1, 2, \ldots$, are the Lanczos vectors generated by the algorithm in section 2.5. Then, following the algorithm of section 2.5, we have

$$\hat{q}_{j+1} = \beta_j^{-1}(\Lambda \hat{q}_j - \alpha_j \hat{q}_j - \beta_{j-1} \hat{q}_{j-1}), \tag{4.23}$$

where

$$\alpha_j = \langle \Lambda \hat{q}_j - \beta_{j-1} \hat{q}_{j-1}, \hat{q}_j \rangle, \quad \beta_j = \| \Lambda \hat{q}_j - \alpha_j \hat{q}_j - \beta_{j-1} \hat{q}_{j-1} \|.$$

It follows that the ith component of $\hat{q}_{j+1}$ is equal to a certain jth-degree polynomial, say, $\psi_j(z)$, evaluated at λ_i, times the ith component of $\hat{q}_1$. The polynomials $\psi_j(z)$, $j = 1, 2, \ldots$, satisfy

$$\psi_j(z) = \beta_j^{-1}(z\psi_{j-1}(z) - \alpha_j \psi_{j-1}(z) - \beta_{j-1}\psi_{j-2}(z)), \tag{4.24}$$

where $\psi_{-1}(z) \equiv 0$, $\psi_0(z) \equiv 1$. If we define the w-inner product of two polynomials ϕ and ψ by

$$\langle \phi(z), \psi(z) \rangle_w \equiv \sum_{i=1}^{n} \phi(\lambda_i)\psi(\lambda_i)\hat{q}_{i1}^2, \tag{4.25}$$

where $\hat{q}_{i1}$ is the ith component of $\hat{q}_1$, then the coefficients in the Lanczos algorithm are given by

$$\alpha_j = \langle z\psi_{j-1}(z) - \beta_{j-1}\psi_{j-2}(z), \psi_{j-1}(z) \rangle_w, \tag{4.26}$$

$$\beta_j = \| z\psi_{j-1}(z) - \alpha_j \psi_{j-1}(z) - \beta_{j-1}\psi_{j-2}(z) \|_w, \tag{4.27}$$

where $\|\phi(z)\|_w \equiv \langle \phi(z), \phi(z) \rangle_w^{1/2}$.

Equation (4.24) with coefficient formulas (4.26–4.27) defines the orthonormal polynomials for the measure corresponding to the w-inner product in (4.25). It follows from the orthonormality of the Lanczos vectors that these polynomials satisfy $\langle \psi_j(z), \psi_k(z) \rangle_w = \delta_{jk}$.

A perturbation vector f_j in the Lanczos algorithm, due to finite precision arithmetic, corresponds to a perturbation $\hat{f}_j = U^H f_j$ of the same size in (4.23). The finite precision analogue of recurrence (4.24) is

$$\psi_j(z) = \beta_j^{-1}(z\psi_{j-1}(z) - \alpha_j \psi_{j-1}(z) - \beta_{j-1}\psi_{j-2}(z) - \zeta_j(z)), \tag{4.28}$$

where $\zeta_j(\lambda_i)\hat{q}_{i1} = \hat{f}_{ij}$. If we imagine that the coefficient formulas (4.26–4.27) hold *exactly* in finite precision arithmetic, where the functions $\psi_j(z)$ now come from the perturbed recurrence (4.28), we still find that the intended orthogonality relation $\langle \psi_j(z), \psi_k(z) \rangle_w = \delta_{jk}$ may fail completely. (It is reasonable to assume that the coefficient formulas (4.26–4.27) hold exactly, since they can be implemented very accurately and any differences between the exact formulas and the computed values can be included in the perturbation term $\zeta_j(z)$.)

It is possible that some coefficient β_j in a finite precision Lanczos computation will be exactly 0 and that the recurrence will terminate, but this is unlikely. If β_j is not 0, then it is positive because of formula (4.27). It follows from a theorem due to Favard [48] that the recurrence coefficients constructed in a finite precision Lanczos computation are the exact recurrence coefficients for the orthonormal polynomials corresponding to *some* nonnegative measure. That is, if we define $\rho_{-1}(z) \equiv 0$, $\rho_0(z) \equiv 1$, and

$$\rho_j(z) = \beta_j^{-1}(z\rho_{j-1}(z) - \alpha_j\rho_{j-1}(z) - \beta_{j-1}\rho_{j-2}(z)) \tag{4.29}$$

for $j = 1, 2, \ldots$, where α_j and β_j are defined by (4.26–4.28), then we have the following theorem.

THEOREM 4.5.1 (Favard). *If the coefficients β_j in* (4.29) *are all positive and the α_j's are real, then there is a measure $d\omega(z)$ such that*

$$\int \rho_j(z)\rho_k(z)\, d\omega(z) = \delta_{jk}$$

for all $j, k = 0, 1, \ldots, \infty$.

The measure $d\omega(z)$ in Favard's theorem is (substantially) uniquely determined, whereas there are infinitely many measures for which the first k polynomials $\rho_0, \ldots, \rho_{k-1}$ are orthonormal. One such measure—a measure with weights on the eigenvalues of T_{k+1}, the weights being the squared first components of each eigenvector of T_{k+1}—was given in section 4.4.1, and another such measure—a measure with weights on points in tiny intervals about the eigenvalues of A—was given in section 4.4.2. It was also shown in [65] that the weight on each interval is approximately equal to the original weight on the corresponding eigenvalue of A; that is, the squared component of $\hat{q}_1$. Thus, the matrix completion result of section 4.4.2 can also be stated in the following way: when one attempts to construct the first k orthonormal polynomials for a measure corresponding to weights on discrete points using the Lanczos algorithm, what one actually obtains are the first k orthonormal polynomials for a slightly different measure—one in which the weights are smeared out over tiny intervals about the original points. Exactly how the weights are distributed over these intervals depends on exactly what rounding errors occur (not just on their size).

It remains an open question whether the measure defined by Favard's theorem has its support in such tiny intervals (i.e., whether δ in Theorem 4.4.3 can be taken to be small and independent of k). If this is not the case, it might still be possible to show that the measure in Favard's theorem is *tiny* everywhere outside such intervals.

Comments and Additional References.

It should come as no surprise that the Lanczos vectors and tridiagonal matrix can be used for many purposes besides solving linear systems. For example, the eigenvalues of T_k can be taken as approximations to some of the eigenvalues of A. It is given as an exercise to show that the orthogonal polynomials defined in section 4.5 are the characteristic polynomials of the successive tridiagonal matrices generated by the Lanczos algorithm. This interpretation enables one to use known properties of the roots of orthogonal polynomials to describe the eigenvalue approximations. In finite precision arithmetic, the fact that the polynomials (or at least a finite sequence of these polynomials) are orthogonal with respect to a slightly smeared-out version of the original measure helps to explain the nature of eigenvalue approximations generated during a finite precision Lanczos computation. Depending on how the tiny intervals of Theorem 4.4.3 are distributed, the corresponding orthogonal polynomials might have several roots in some of the intervals before having any roots in some of the others. This is usually the case with an interval corresponding to a large well-separated eigenvalue. This explains the observed phenomenon of multiple close approximations to some eigenvalues appearing in finite precision Lanczos computations before any approximations to some of the other eigenvalues appear.

The Lanczos vectors and tridiagonal matrix can also be used very effectively to compute the matrix exponential $\exp(tA)\varphi$, which is the solution at time t to the system of differential equations $y' = Ay$, $y(0) = \varphi$. Similar arguments to those used here show why the nonorthogonal vectors generated by a finite precision Lanczos computation can still be used effectively for this purpose [34]. For a number of other applications, including discussions of the effects of finite precision arithmetic, see, for example, [35, 60].

The effect of rounding errors on the CG algorithm has been a subject of concern since the algorithm was first introduced in 1952 by Hestenes and Stiefel [79]. It was recognized at that time that the algorithm did not always behave the way exact arithmetic theory predicted. For example, Engeli et al. [43] applied the CG method (without a preconditioner) to the biharmonic equation and observed that convergence did not occur until well after step n. For this and other reasons, the algorithm did not gain widespread popularity at that time.

With the idea of preconditioning in the CG method, interest in this algorithm was revived in the early 1970's [115, 27], and it quickly became the method of choice for computations involving large Hermitian positive definite matrices. Whatever the effect of roundoff, it was observed that the method performed very well in comparison to other iterative methods. Further attempts were made to explain the success of the method, mostly using the interpretation given in section 2.3 that the algorithm minimizes the A-norm of the error in a plane that includes the direction of steepest descent. Using this argument, Wozniakowski [143] showed that a special version of the CG algorithm does, indeed, reduce the A-norm of the error at each step by at least as much as a steepest descent step, even in finite precision arithmetic. Cullum and Willoughby [30] proved a similar result for a more standard version of the algorithm. Still, a more global approach was needed to explain why the CG algorithm converges so much faster than the method of steepest descent; e.g., it converges at least as fast as the Chebyshev algorithm. Paige's work on the Lanczos algorithm [109] provided a key in this direction. A number of analyses were developed to explain the behavior of the CG algorithm using information from the entire computation (i.e., the matrix equation (2.23)), instead of just one or two steps (e.g., [35, 62, 65, 121]). The analogy developed in this chapter, identifying the finite precision computation with the exact algorithm applied to a different matrix, appears to be very effective in explaining and predicting the behavior of the CG algorithm in finite precision arithmetic [71]. The numerical examples presented in section 4.1 were first presented in [126].

Exercises.

4.1. Show that the orthonormal polynomials defined by (4.24) are the *characteristic polynomials* of the tridiagonal matrices generated by the Lanczos algorithm.

4.2. How must the error bound you derived in Exercise 3.1 for a matrix with a small, well-separated eigenvalue be modified for finite precision arithmetic? Does the finite precision error bound differ more from that of exact arithmetic in the case when a positive definite coefficient matrix has one eigenvalue much smaller than the others or in the case when it has one eigenvalue much larger than the others? (This comparison can be used to explain why one preconditioner might be considered better based on exact arithmetic theory, but a different preconditioner might perform better in actual computations. See [133] for a comparison of incomplete Cholesky and modified incomplete Cholesky decompositions, which will be discussed in Chapter 11.)