SIAM J. MATRIX ANAL. APPL.
Vol. 31, No. 5, pp. 2347–2359

AN AUGMENTED STABILITY RESULT FOR THE LANCZOS HERMITIAN MATRIX TRIDIAGONALIZATION PROCESS*

CHRISTOPHER C. PAIGE†

Abstract. It is shown that a good implementation of the Hermitian matrix tridiagonalization process of Lanczos [*J. Research Nat. Bur. Standards*, 45 (1950), pp. 255–282] produces a tridiagonal matrix that is, at each step, the exact result for the process applied to a strange augmented problem. Since the process is not stable in the standard sense, this augmented stability result cannot be transformed to prove standard stability. The intent is to obtain an increased understanding of the Lanczos tridiagonalization process, and this result could later be used to analyze the many applications of the process to large sparse matrix problems, such as the solution of the eigenproblem, compatible linear systems, least squares, and the singular value decomposition.

Key words. loss of orthogonality, rounding error analysis, Lanczos process, Lanczos tridiagonalization, Hermitian matrix tridiagonalization, large sparse matrix computations

AMS subject classifications. 65F10, 65F25, 65F30, 65F50, 65G50, 15A23, 15A57

DOI. 10.1137/090761343

1. Introduction.

Here we will use "the Lanczos process" to mean the famous Hermitian matrix tridiagonalization process of Lanczos [15]:

$$AV_k = V_kT_k + v_{k+1}\beta_{k+1}e_k^T = V_{k+1}T_{k+1,k}, \quad V_{k+1}^HV_{k+1} = I_{k+1}, \;\; T_k \text{ tridiagonal.} \tag{1.1}$$

In theory the Lanczos process produces a sequence of orthonormal n-vectors, which are columns of $V_k = [v_1, \ldots, v_k] \in \mathbb{C}^{n\times k}$, from a given unit length (i.e., 2-norm of 1) n-vector v_1 via a sequence of matrix-vector multiplications with the given Hermitian matrix $A \in \mathbb{C}^{n\times n}$. These vectors are obtained by the orthogonalization of each successively produced vector against the two previously computed orthonormal vectors, followed by the normalization of the resulting orthogonal vector. With finite precision computation this algorithm produces a sequence of n-vectors which can have a severe loss of orthogonality, but where each vector has a 2-norm that is almost 1. Here it is shown that a good implementation of the Lanczos process produces a tridiagonal matrix T_k that is exact for a strange augmented problem. Since the augmented problem differs in a significant way from the original matrix A (but of course includes A), we will *not* say that the Lanczos process is "augmented backward stable." We have not yet decided on a satisfactory nomenclature, so we will for the moment refer to it as the "strange augmented stability," or just "augmented stability," of the process. The intent of this analysis is to obtain an increased understanding of the Lanczos process and its practical use for large sparse matrix problems such as the eigenproblem, solution of linear systems and least squares, singular value computations, and related problems; see, for example, [2, 7, 13, 15, 16, 26, 27, 29], and also [28, section 3] for comments by Saunders on regularization and partial least squares.

*Received by the editors June 8, 2009; accepted for publication (in revised form) by M. H. Gutknecht April 20, 2010; published electronically July 13, 2010. This work was supported by NSERC of Canada grant OGP0009236.
http://www.siam.org/journals/simax/31-5/76134.html
†School of Computer Science, McGill University, Montreal, Quebec, H3A 2A7, Canada (paige@cs.mcgill.ca).

Lanczos originally presented his tridiagonalization process in [15] for solving the eigenproblem, but mentioned it would be useful for solution of equations, and in [16] he adapted it for this purpose when the matrix is symmetric positive definite. This latter method is mathematically equivalent to Hestenes and Stiefel's method of conjugate gradients (CG) in [13]. The Lanczos process applied to the real symmetric matrix eigenproblem was soon superseded by the backward stable method of Givens [6] based on matrix factorizations (see [36, Chap. 5, sections 22–35, pp. 282–299]), while CG fell out of favor. Later both methods were found to be advantageous for many types of large sparse matrix problems; see, for example, [20, 31]. Today the Lanczos process is the basis for several methods which are still considered among the best we have for large sparse matrix problems (see, for example, [5, 17]), and for this reason alone it is important to understand its strange numerical behavior as deeply as possible.

The initial analysis of this behavior appeared in [20, 21, 22, 23]. This was taken up by Parlett and several of his students, who greatly improved the use and understanding of the process. See, for example, [30] for helpful clarifications and explanations of many of the important ideas and relations. Greenbaum, and independently Strakoš, developed our understanding of the practical behavior of the Lanczos process and its use for both the eigenproblem and CG; see, for example, [9, 10, 11, 34, 35]. Many others also contributed to the understanding of the subtle behavior of these algorithms; see, for example, Wülling [37, 38] and Zemke [39, 40] for some recent research in the area. For a full history and description of these developments until recently, see the text by Meurant [17]. An elegant approach to some of the important theory and practical behavior of both the Lanczos process and CG, together with a good historical outline, is given by Meurant and Strakoš in [18].

An augmented result on the stability of the Lanczos process was given by Greenbaum in [10]. Corollary 3.2 here gives a result of similar tenor, and we compare these two results after Corollary 3.3. Following this work of Greenbaum, and the orthogonal polynomial and Gauss quadrature relationships described in [10, 13] and elsewhere, Strakoš and coworkers have developed illuminating results on the practical behaviors of the Lanczos process and CG via an analysis based on the fundamental relationship with the theory of orthogonal polynomials and Gauss quadrature of the Riemann–Stieltjes integral; see the survey paper [18] for a nice description, and [19] for further developments and an extensive literature survey.

The approach here has led to some similar results for the real symmetric eigenproblem, but it is instead based purely on ideas from matrix theory and an extension of the concept of backward stability for numerical algorithms introduced by Wilkinson, whose work motivated the work here so strongly; see, for example, [14, 36]. The results obtained so far with this direct approach complement the understanding gained by those earlier approaches.

In section 2 we state a theorem from [24, Theorem 2.1] on how a particular $(n+k)\times(n+k)$ unitary matrix $Q^{(k)}$ can be derived from any $n\times k$ matrix V_k whose columns have 2-norms of one. We will use this with basic rounding error results to prove the strange augmented stability of the Lanczos process in section 3.

Since the Lanczos process is not backward stable in the standard sense, this augmented stability result cannot be transformed to prove standard backward stability of the Lanczos process. However, it has been designed to be used to obtain more standard results for the many applications of the Lanczos process, and here we give a few words at the end of section 3.3 regarding the eigenproblem.

The extension of Theorem 2.1 to handle biorthogonal sets of vectors in [24, Theorem 7.1] suggests that some of the results here might also be extended to some variant of the Lanczos unsymmetric matrix tridiagonalization process in [15]; see also, for example, [36, Chap. 6, sections 35–40, pp. 388–394].

A bit more history will add another reason why we do not attempt to provide any more general results than proving the strange augmented stability of the Lanczos process. Perhaps the first augmented backward stability result was that initiated by Sheffield [33] on the augmented backward stability of the so-called modified Gram–Schmidt algorithm (MGS); see [3, equation (3.3)]. This was used by Björck and Paige [3, 4] to analyze and suggest improved ways of using MGS for least squares and related problems. The same idea was used in [25] to show the backward stability of the MGS-GMRES algorithm in [32] for the solution of linear equations. Since MGS orthogonalizes against all previous vectors, it was possible in these cases to transform the augmented results to standard results using, for example, [3, Lemma 3.1], which was later improved slightly in [24, Theorem 4.1].

Barlow, Bosner, and Drmač [1] used Sheffield's insight to prove some numerical stability properties of their algorithm for the bidiagonalization of a matrix by orthogonal transformations from the left and right. Their method used Householder transformations to produce the effect of the smaller-dimensioned orthogonal matrix, and this forced finite termination. But it used local orthogonalization by vector subtraction to find the columns of the larger-dimensioned orthogonal matrix. This led to a saving in floating point operations, often at the cost of significant loss of orthogonality in the latter's columns. It was shown in [24] how Theorem 2.1 here could be used to give a simpler and shorter rounding error analysis of their algorithm. Then the finite termination property of their algorithm made it possible to obtain standard results directly from augmented results. But because of loss of orthogonality in practice, the Lanczos process has no finite termination property—it can go on forever. This means that the computed tridiagonal matrix can have a greater dimension than the original Hermitian matrix, and as we will show, the augmented result is startlingly different. In general it is not straightforward to obtain results in standard form, and it will be necessary to treat each of the applications of the Lanczos process individually. So only the essential augmented stability result will be given here.

1.1. Notation. We will use "$\triangleq$" for "is defined to be" and "$\equiv$" for "is equivalent to." We will say a complex nonsquare $n \times k$ matrix Q_1 has orthonormal columns if $Q_1^H Q_1 = I$ and write $Q_1 \in \mathcal{U}^{n\times k}$, while Q_1 and Q_2 are orthogonal if $Q_1^H Q_2 = 0$. For floating point arithmetic our measure of relative precision will be the *unit roundoff* (see, e.g., [14]) and will be denoted by ϵ. I_n denotes the $n \times n$ unit matrix (but we will sometimes use I), e_j will be the jth column of a unit matrix I, so Be_j is the jth column of B, while e will be a vector of 1s of the required dimension. We will use $\sigma(\cdot)$ to denote a singular value and define $\kappa_2(B) \triangleq \sigma_{max}(B)/\sigma_{min}(B)$. We will denote the absolute value of a matrix B by $|B|$, the Frobenius norm by $\|B\|_F \triangleq \sqrt{\mathrm{trace}(B^H B)}$, the vector 2-norm by $\|v\|_2 \triangleq \sqrt{v^H v}$, and its subordinate matrix norm by $\|B\|_2 \triangleq \sigma_{max}(B)$.

The matrices E (whose columns are Ee_j, not e_j), F, G, and H will denote small terms introduced by rounding errors. For the rounding error analysis we will use a simplistic notation such as $\|E_k\|_{2,F} \leq O(\epsilon)\|A\|_2$ to denote bounds for the basic error terms E_k in (3.2) and F_k in (3.5). This is in order to accommodate the various possible bounds like those in section 3.2 that have been, and may yet be, found for these. This precludes the more precise notation used in [14, pp. 63–68].

We will usually index matrices by subscripts as in V_k when the $(k+1)$st matrix can be obtained from the kth by adding a column, or a column and a row. Otherwise we will use superscripts, as in $H^{(k)}$. We will partition $Q^{(k)} = [Q_1^{(k)}, Q_2^{(k)}]$.

We use SUT to mean "strictly upper triangular," while "sut($\cdot$)" gives the matrix in parentheses with its lower triangle set to zero; thus $\mathrm{sut}(\alpha) = 0$ for a scalar α. Similarly SLT means "strictly lower triangular," LT means "lower triangular," and "lt($\cdot$)" gives the matrix in parentheses with its SUT part set to zero.

2. Obtaining a unitary matrix from unit 2-norm *n*-vectors. A crucial tool used in this paper is a theorem which was proved in [24], which we restate here for convenience. It allows us to develop an $(n+k)\times(n+k)$ unitary matrix $Q^{(k)}$ from any $n\times k$ matrix V_k with unit 2-norm columns. When V_k comes from the Lanczos process, this allows us to obtain our strange augmented stability result from earlier results in this area.

THEOREM 2.1. *For any integers $n \geq 1$ and $k \geq 1$, and $V_k \triangleq [v_1, \ldots, v_k] \in \mathbb{C}^{n\times k}$ with $\|v_j\|_2 = 1$, $j = 1, \ldots, k$, define the strictly upper triangular matrix S_k as follows:*

$$S_k \triangleq (I_k + U_k)^{-1}U_k \equiv U_k(I_k + U_k)^{-1} \in \mathbb{C}^{k\times k}, \qquad U_k \triangleq \mathrm{sut}(V_k^H V_k) \tag{2.1}$$

(where clearly $I_k \pm S_k$ and $I_k \pm U_k$ are always nonsingular). Then

$$U_k S_k = S_k U_k, \quad U_k = (I_k - S_k)^{-1}S_k \equiv S_k(I_k - S_k)^{-1}, \quad (I_k - S_k)^{-1} = I_k + U_k, \tag{2.2}$$

$$(I_k - S_k)^H V_k^H V_k (I_k - S_k) = I_k - S_k^H S_k, \tag{2.3}$$

$$(I_k - S_k) V_k^H V_k (I_k - S_k)^H = I_k - S_k S_k^H, \tag{2.4}$$

$$\|S_k\|_2 \leq 1; \quad V_k^H V_k = I \Leftrightarrow \|S_k\|_2 = 0; \quad V_k^H V_k \ \textit{singular} \Leftrightarrow \|S_k\|_2 = 1. \tag{2.5}$$

Most importantly, S_k is the unique *strictly upper triangular $k\times k$ matrix such that*

$$Q^{(k)} \triangleq \left[Q_1^{(k)} \;\middle|\; Q_2^{(k)} \right] \triangleq \left[\begin{array}{c|c} S_k & (I_k - S_k)V_k^H \\ V_k(I_k - S_k) & I_n - V_k(I_k - S_k)V_k^H \end{array} \right] \in \mathcal{U}^{(n+k)\times(n+k)}. \tag{2.6}$$

If we write $\left[\begin{smallmatrix} \widehat{S}_k & s_{k+1} \\ 0 & 0 \end{smallmatrix}\right] \triangleq S_{k+1}$, then we also have $\widehat{S}_k = S_k$ and

$$s_{k+1} = (I_k - S_k)V_k^H v_{k+1}, \quad \begin{bmatrix} S_{k+1} \\ V_{k+1}(I_{k+1} - S_{k+1}) \end{bmatrix} = \left[\begin{array}{c|c} S_k & s_{k+1} \\ 0 & 0 \\ V_k(I_k - S_k) & v_{k+1} - V_k s_{k+1} \end{array} \right]. \tag{2.7}$$

Here we add a simple consequence of (2.1), and a generalization of (2.5).

COROLLARY 2.2. *With the notation in Theorem* 2.1,

$$s_{j-1,j} \triangleq e_{j-1}^T S_k e_j = u_{j-1,j} \triangleq e_{j-1}^T U_k e_j = v_{j-1}^H v_j, \quad j = 2, \ldots, k; \tag{2.8}$$

$$k = \mathrm{rank}(V_k) + \textit{the number of unit singular values of } S_k. \tag{2.9}$$

Proof. First, (2.8) follows from the $(j–1, j)$ element of $(I + U_k)S_k = U_k$; see (2.1). Let the eigenvalue decomposition of Hermitian nonnegative definite (2.3) be

$$P^H(I - S_k)^H V_k^H V_k (I - S_k)P = P^H(I - S_k^H S_k)P = \mathrm{diag}(O_d, \Gamma), \quad P^H = P^{-1},$$

O_d being the $d\times d$ zero matrix, and $(k-d)\times(k-d)$ diagonal Γ having positive diagonal elements. Then $V_k(I - S_k)$, and so V_k, has rank $k - d$. But

$$P^H S_k^H S_k P = I - \mathrm{diag}(O_d, \Gamma) = \mathrm{diag}(I_d, I - \Gamma),$$

and so S_k has exactly d unit singular values, proving (2.9). ☐

We see from (2.9) that if $k > n$, then S_k has at least $k - n$ unit singular values.

One important result of the theorem is that $\|S_k\|_2$ is an excellent measure of the loss of orthogonality in the (unit length) columns of V_k; see [25, Lemma 5.1] and [24, Corollary 5.2]. From these, for V_k, U_k, and S_k in Theorem 2.1,

$$1-2\|U_k\|_2 \le \frac{1-\|S_k\|_2}{1+\|S_k\|_2} \le \sigma_i^2(V_k) \le 1+2\|U_k\|_2 \le \frac{1+\|S_k\|_2}{1-\|S_k\|_2}, \tag{2.10}$$

$$\sigma_{min}(V_k) \le 1 \le \sigma_{max}(V_k); \qquad \sigma_{min}^{-2}(V_k) \;\text{ and }\; \kappa_2(V_k) \le \frac{1+\|S_k\|_2}{1-\|S_k\|_2}. \tag{2.11}$$

Theorem 2.1 states that U_k and S_k, $k>1$, are obtained from U_{k-1} and S_{k-1} by adding a column and a row, so from (2.6) and (2.7) this is true for $Q_1^{(k)}$ too. Nevertheless we write $Q_1^{(k)}$ because it is part of $Q^{(k)} = \left[Q_1^{(k)} \mid Q_2^{(k)}\right]$. The column and row added to S_{k-1} to give S_k are as follows. Write $R_k \triangleq I + U_k = \left[\begin{smallmatrix} R_{k-1} & u_k \\ 0 & 1\end{smallmatrix}\right]$, where $u_k \triangleq V_{k-1}^H v_k$ (see (2.1)) so $R_k^{-1} = \left[\begin{smallmatrix} R_{k-1}^{-1} & -R_{k-1}^{-1}u_k \\ 0 & 1\end{smallmatrix}\right]$, and then we can use the subscript indexing notation for S_k, since from (2.1)

$$S_k \triangleq R_k^{-1}U_k = \begin{bmatrix} R_{k-1}^{-1} & -R_{k-1}^{-1}u_k \\ 0 & 1\end{bmatrix}\begin{bmatrix} U_{k-1} & u_k \\ 0 & 0\end{bmatrix} = \begin{bmatrix} S_{k-1} & s_k \\ 0 & 0\end{bmatrix}, \quad s_k \triangleq R_{k-1}^{-1}u_k.$$

In [24] the construction in Theorem 2.1 was called a *unitary or orthonormal augmentation of an array or sequence of unit length vectors* (the "augmentation" from V_k to $Q_1^{(k)}$ in (2.6)). It was thought to be useful in the rounding error analysis of any algorithm that produces a sequence of orthonormal vectors, but because of rounding errors fails to do so to a significant extent. The present paper describes a very basic, and perhaps the most important, possible use—a rounding error analysis of the Lanczos process.

3. Application to the Lanczos algorithm.

Throughout this section we will assume $\beta_2\beta_3\cdots\beta_{k+1} \ne 0$. This almost always happens in practice, and we would stop at the first zero β_j if it did not. For generality we will consider the complex case.

3.1. Basic rounding error results.

Let the columns of $\widetilde{V}_k \triangleq [\tilde v_1,\dots,\tilde v_k]$ be the first k vectors obtained by using a reliable implementation (such as that in section 3.2) of the Lanczos process with the $n\times n$ Hermitian matrix A, and define

$$V_k \triangleq [v_1,\dots,v_k] \triangleq \widetilde{V}_k\widetilde{D}_k^{-1}, \quad \widetilde{D}_k \triangleq \operatorname{diag}(\|\tilde v_j\|_2) \quad \text{giving } \|v_j\|_2 = 1,\ j=1,\dots,k. \tag{3.1}$$

After k steps of the Lanczos algorithm with unit roundoff ϵ, we have (see, e.g., [22])

$$T_{k+1,k} \triangleq \begin{bmatrix} T_k \\ \beta_{k+1}e_k^T\end{bmatrix} \triangleq \begin{bmatrix} \alpha_1 & \beta_2 & & \\ \beta_2 & \alpha_2 & \beta_3 & \\ & \cdot & \cdot & \cdot \\ & & \beta_k & \alpha_k \\ & & & \beta_{k+1}\end{bmatrix},$$

$$AV_k = V_kT_k + v_{k+1}\beta_{k+1}e_k^T + E_k = V_{k+1}T_{k+1,k} + E_k, \tag{3.2}$$

$\|E_k\|_{2,F} \le O(\epsilon)\|A\|_2$ in [22]. We write E_k rather than $E^{(k)}$ since $E_k = [E_{k-1}, E_ke_k]$.

Let $U_k \triangleq \operatorname{sut}(V_k^HV_k)$, $u_{k+1} \triangleq V_k^Hv_{k+1}$, $u_{ij} \triangleq v_i^Hv_j$; then from symmetry

$$\begin{aligned} V_k^HAV_k &= (U_k^H + I + U_k)T_k + u_{k+1}\beta_{k+1}e_k^T + V_k^HE_k \\ &= T_k(U_k^H + I + U_k) + e_k\beta_{k+1}u_{k+1}^H + E_k^HV_k. \end{aligned} \tag{3.3}$$

Equating the upper triangular parts in this last equality shows that

$$T_kU_k - [U_k, u_{k+1}]T_{k+1,k} = F_k, \qquad F_k \triangleq D_k + \mathrm{sut}(V_k^H E_k - E_k^H V_k), \\ D_k \triangleq \mathrm{diag}(-u_{12}\beta_2, u_{12}\beta_2 - u_{23}\beta_3, \ldots, u_{k-1,k}\beta_k - u_{k,k+1}\beta_{k+1}), \tag{3.4}$$

and when a good algorithm has been used we have (see, for example, section 3.2)

$$\|F_k\|_{2,F} \le O(\epsilon)\|A\|_2. \tag{3.5}$$

We write F_k rather than $F^{(k)}$, and note that $F_k + E_k^H V_k$ is Hermitian, since

$$F_k = \left[\begin{array}{c|c} \begin{matrix} F_{k-1} \\ 0 \end{matrix} & F_k e_k \end{array} \right], \qquad F_k + E_k^H V_k = D_k + \mathrm{sut}(V_k^H E_k) + \mathrm{lt}(E_k^H V_k). \tag{3.6}$$

There are different bounds for different situations. Here we will give our results in terms of E_k and F_k, so that anyone can include their own bounds on these.

3.2. Possible rounding error bounds. Our main results will be independent of the particular bounds, but to give a feeling for the context, we give one example of the bounds on E_k and F_k in (3.2) and (3.4). According to [17, p. 96] the most used variant of the real symmetric Lanczos algorithm is that recommended in [23, section 2] (but care should be taken to ensure real α_j in the Hermitian case): For a given $b \neq 0$,

$$\beta := +(b^H b)^{\frac{1}{2}}, \quad v_1 := b/\beta, \quad w := Av_1. \quad \text{For } \ j = 1, 2, \ldots, k \ \text{ repeat the following:}$$
$$\begin{cases} \alpha_j := v_j^H w, \quad w := w - v_j\alpha_j, \quad \beta_{j+1} := +(w^H w)^{\frac{1}{2}}, \\ \text{if } \quad \beta_{j+1} = 0, \quad \text{then STOP, else: } v_{j+1} := w/\beta_{j+1}, \quad w := Av_{j+1} - v_j\beta_{j+1}. \end{cases}$$

It is relevant to note that this is a two two-term Krylov process ($w := Av_j - v_{j-1}\beta_j$ and $w := w - v_j\alpha_j$) rather than the one three-term process $v_{j+1}\beta_{j+1} = Av_j - v_j\alpha_j - v_{j-1}\beta_j$, and has some advantages similar to the shorter recurrences discussed by Gutknecht and Strakoš in [12]. The bounds below were obtained for the computed v_j, not for their correctly normalized versions; however, the differences will be minimal.

Here is an example of bounds for the real case outlined in [17, Chap. 3]. If A has at most m nonzero elements in any row, then with the definitions and restrictions

$$\alpha \triangleq \||A|\|_2/\|A\|_2, \quad \epsilon_0 \triangleq 2(n+4)\epsilon < 1/12, \quad \epsilon_1 \triangleq 2(7 + m\alpha)\epsilon, \quad k(3\epsilon_0 + \epsilon_1) \le 1,$$

it was shown in [22] (see also [23, section 2] and [17, section 3.3]) that with the above algorithm for $j = 1, 2, \ldots, k$ the error terms E_k and F_k satisfy

$$\|E_k\|_2 \le \|E_k\|_F \le k^{\frac{1}{2}}\epsilon_1\|A\|_2, \qquad \|F_k\|_2 \le \|F_k\|_F \le \sqrt{2k(k\epsilon_1^2 + 8\epsilon_0^2)}\|A\|_2. \tag{3.7}$$

Of course these are just bounds (probably quite weak ones), and actual values will tend to be far smaller.

3.3. The nearby problem. We showed in Theorem 2.1 that if we carried out the orthonormal augmentation of V_k, then we obtained (see (2.6) and (2.2)) $\mathrm{sut}(V_k^H V_k) = U_k = S_k(I - S_k)^{-1} = (I - S_k)^{-1}S_k$, where S_k was SUT and

$$S_{k+1} = \begin{bmatrix} S_k & s_{k+1} \\ 0 & 0 \end{bmatrix}, \qquad s_{k+1} = (I - S_k)V_k^H v_{k+1} = (I - S_k)u_{k+1}; \tag{3.8}$$

see (2.7). Using the fact that $e_k^T S_k = 0$ since S_k is SUT, we see that

$$S_k T_k S_k = [S_k T_k + s_{k+1}\beta_{k+1}e_k^T]S_k = [S_k, s_{k+1}]T_{k+1,k}S_k,$$

so that we have for the upper triangular (3.4)

$$\begin{aligned} &T_k S_k (I - S_k)^{-1} - [(I - S_k)^{-1}S_k, (I - S_k)^{-1}s_{k+1}]T_{k+1,k} = F_k,\\ &(I - S_k)F_k(I - S_k) = (I - S_k)T_k S_k - [S_k, s_{k+1}]T_{k+1,k}(I - S_k). \end{aligned}$$

This gives the following two different forms of the one useful result:

$$T_k S_k - [S_k, s_{k+1}]T_{k+1,k} = T_k S_k - S_k T_k - s_{k+1}\beta_{k+1}e_k^T = (I - S_k)F_k(I - S_k), \tag{3.9}$$

$$(I - S_k)T_k = T_k(I - S_k) + s_{k+1}\beta_{k+1}e_k^T + (I - S_k)F_k(I - S_k). \tag{3.10}$$

It will be seen that these different forms help in different places. It was mentioned in [24] that for Theorem 2.1 to be useful in a rounding error analysis, an important ancillary result will be an expression for S_k. Here (3.9), or (3.10), is the key expression for S_k. Note that (3.4) for U_k and (3.9) for S_k have equivalent forms on the left-hand side. In [20] it was hoped that (3.4) would reveal all, but this paper indicates that we also need (3.9), or the mathematically equivalent (3.10).

We will need the following simple results. Using (3.3) and (3.4),

$$\begin{aligned} V_k^H A V_k &= U_k^H T_k + T_k + [U_k, u_{k+1}]T_{k+1,k} + V_k^H E_k\\ &= U_k^H T_k + T_k + T_k U_k + (V_k^H E_k - F_k). \end{aligned} \tag{3.11}$$

From (3.2) and the fact that $e_k^T S_k = 0$,

$$A V_k (I - S_k)V_k^H = (V_k T_k + E_k)(I - S_k)V_k^H + v_{k+1}\beta_{k+1}v_k^H, \tag{3.12}$$

while it is obvious that

$$(I - S_k)^H T_k (I - S_k) - T_k(I - S_k) - (I - S_k)^H T_k = S_k^H T_k S_k - T_k. \tag{3.13}$$

Note from (2.2) that $(I - S_k)^{-1}e_k = e_k + U_k e_k = V_k^H v_k$, so with (3.8)

$$(I - S_k)V_k^H v_k = e_k, \qquad (I - S_k)V_k^H v_{k+1} = s_{k+1}. \tag{3.14}$$

The ideal Lanczos process (1.1) corresponds to the (partial) unitary similarity transformation of A to tridiagonal form, so that $V_k^H A V_k = T_k$ up to $k = n$. We now show in (3.15) that, even with severe loss of orthogonality, a correctly programmed *computational* process also corresponds to an *exact* unitary similarity transformation of a matrix involving A into a developing tridiagonal form with the *computed* T_k.

THEOREM 3.1. *After k finite precision steps of a Lanczos algorithm with $A = A^H$ and v_1, $\|v_1\|_2 = 1$, leading to V_{k+1} with unit-norm columns (see* (3.1)*), β_{k+1} and T_k in section* 3.1*, with S_k, s_{k+1}, and $Q^{(k)}$ defined in Theorem* 2.1*, E_k and F_k satisfying*

(3.2) *and* (3.4), *and* $A_k \triangleq A - v_{k+1}\beta_{k+1}v_k^H - v_k\beta_{k+1}v_{k+1}^H = A_k^H$, *we have*

$$Q^{(k)H}\left(\begin{bmatrix} T_k & 0 \\ 0 & A \end{bmatrix} + H^{(k)}\right)Q^{(k)} = \left[\begin{array}{c|c} T_k & e_k\beta_{k+1}v_{k+1}^H \\ \hline v_{k+1}\beta_{k+1}e_k^T & A_k \end{array}\right], \tag{3.15}$$

$$Q^{(k)} \triangleq \left[Q_1^{(k)} \;\middle|\; Q_2^{(k)}\right] \triangleq \left[\begin{array}{c|c} S_k & (I-S_k)V_k^H \\ V_k(I-S_k) & I-V_k(I-S_k)V_k^H \end{array}\right], \quad Q^{(k)H} = \{Q^{(k)}\}^{-1}, \tag{3.16}$$

$$H^{(k)} \triangleq N_k(F_k + E_k^H V_k)N_k^H + \begin{bmatrix} 0 \\ E_k \end{bmatrix} N_k^H + N_k\begin{bmatrix} 0 & E_k^H \end{bmatrix} = H^{(k)H}, \tag{3.17}$$

$$N_k \triangleq \begin{bmatrix} I_k \\ -V_k \end{bmatrix}(I_k - S_k) = \begin{bmatrix} I_k \\ 0 \end{bmatrix} - Q_1^{(k)}, \qquad \|N_k\|_2 \leq 2, \tag{3.18}$$

$$\|H^{(k)}\|_{2,F} \leq 4\,(\|E_k\|_{2,F} + \|F_k\|_{2,F}), \tag{3.19}$$

where the 2-*, or* F-*, norm is to be used consistently throughout this last inequality.*

Proof. We see that (3.16) follows from Theorem 2.1. Then $F_k + E_k^H V_k$ is Hermitian from (3.6), and so the expression for $H^{(k)}$ is Hermitian in (3.17), as it should be for (3.15) to hold. Also the equality in (3.18) follows from (3.16), and then $\|N_k\|_2^2 = \|N_k^H N_k\|_2 = \|2I - S_k - S_k^H\|_2 \leq 4$; see (2.5). The proof of the remaining results will follow by obtaining expressions for the subblocks of G defined below. These blocks will be small, resulting in small $\|H^{(k)}\|_{2,F}$, when $\|E_k\|_{2,F}$ and $\|F_k\|_{2,F}$ are small,

$$G \triangleq \left[\begin{array}{c|c} G_{1,1} & G_{1,2} \\ \hline G_{2,1} & G_{2,2} \end{array}\right] \triangleq \left[\begin{array}{c|c} T_k & 0 \\ \hline 0 & A \end{array}\right] Q^{(k)} - Q^{(k)} \left[\begin{array}{c|c} T_k & e_k\beta_{k+1}v_{k+1}^H \\ \hline v_{k+1}\beta_{k+1}e_k^T & A_k \end{array}\right]. \tag{3.20}$$

To obtain an expression for G we use (3.20) with (3.16). To make this readable in an acceptable amount of space, we temporarily make the substitutions

$$T \equiv T_k, \quad S \equiv S_k, \quad V \equiv V_k, \quad v \equiv v_{k+1}, \quad s \equiv s_{k+1}, \quad \beta \equiv \beta_{k+1}, \quad E \equiv E_k, \quad F \equiv F_k.$$

Then for $G_{1,1}$ in (3.20) we see from (3.16), (3.14), and (3.9) that

$$G_{1,1} = TS - ST - (I-S)V^H v\beta e_k^T = TS - ST - s\beta e_k^T = (I-S)F(I-S). \tag{3.21}$$

Next, from (3.20) and (3.16), with (3.10), (3.14), and (3.2),

$$\begin{aligned} G_{2,1} &= AV(I-S) - V(I-S)T - v\beta e_k^T + V(I-S)V^H v\beta e_k^T \\ &= AV(I-S) - VT(I-S) - Vs\beta e_k^T - V(I-S)F(I-S) - v\beta e_k^T + Vs\beta e_k^T \\ &= (AV - VT - v\beta e_k^T)(I-S) - V(I-S)F(I-S) = [E - V(I-S)F](I-S). \end{aligned} \tag{3.22}$$

Then from (3.20) and (3.16), with (3.10), (3.14), and (3.2)

$$\begin{aligned} G_{1,2} &= T(I-S)V^H - Se_k\beta v^H - (I-S)V^H A + (I-S)V^H v\beta v_k^H + (I-S)V^H v_k\beta v^H \\ &= (I-S)TV^H - s\beta v_k^H - (I-S)F(I-S)V^H - Se_k\beta v^H - (I-S)V^H A + s\beta v_k^H + e_k\beta v^H \\ &= (I-S)(V^H A - E^H - e_k\beta v^H) - (I-S)F(I-S)V^H + (I-S)e_k\beta v^H - (I-S)V^H A \\ &= -(I-S)[E^H + F(I-S)V^H]. \end{aligned}$$

Finally, from (3.20) and (3.16), with (3.14), (3.2), and (3.10),

$$\begin{aligned}G_{2,2} = {} & A - AV(I-S)V^H - V(I-S)e_k\beta v^H - A + v\beta v_k^H + v_k\beta v^H \\ & + V(I-S)V^H A - V(I-S)V^H v\beta v_k^H - V(I-S)V^H v_k\beta v^H \\ = {} & V(I-S)V^H A - AV(I-S)V^H - V(I-S)e_k\beta v^H \\ & + v\beta v_k^H + v_k\beta v^H - Vs\beta v_k^H - v_k\beta v^H \\ = {} & V(I-S)(TV^H + e_k\beta v^H + E^H) - (VT + v\beta e_k^T + E)(I-S)V^H \\ & - V(I-S)e_k\beta v^H + v\beta v_k^H - Vs\beta v_k^H \\ = {} & V(I-S)(TV^H + E^H) - (VT + E)(I-S)V^H - Vs\beta v_k^H \\ = {} & V[(I-S)T - T(I-S) - s\beta e_k^T]V^H + V(I-S)E^H - E(I-S)V^H \\ = {} & V(I-S)F(I-S)V^H + V(I-S)E^H - E(I-S)V^H.\end{aligned}$$

Combining these submatrix expressions and rewriting as a sum of factors gives

$$\begin{aligned}G &= \left[\begin{array}{c|c} (I-S)F(I-S) & -(I-S)[E^H + F(I-S)V^H] \\ \hline [E-V(I-S)F](I-S) & V(I-S)F(I-S)V^H + V(I-S)E^H - E(I-S)V^H \end{array}\right] \\ &= \begin{bmatrix} I \\ -V \end{bmatrix}(I-S)F(I-S)\left[I \,\middle|\, -V^H\right] + \begin{bmatrix} 0 \\ E \end{bmatrix}(I-S)\left[I \,\middle|\, -V^H\right] - \begin{bmatrix} I \\ -V \end{bmatrix}(I-S)\left[0 \,\middle|\, E^H\right].\end{aligned}$$

But from (3.15) and (3.20) $H^{(k)} = -GQ^{(k)H}$, where from (3.16) and (3.18)

$$\begin{aligned}&(I-S)\left[I \,\middle|\, -V^H\right] = \left[I-S \,\middle|\, -(I-S)V^H\right] = \left[I_k \,\middle|\, 0\right](I - Q^{(k)}), \\ &(I-S)\left[I \,\middle|\, -V^H\right]Q^{(k)H} = \left[I_k \,\middle|\, 0\right](Q^{(k)H} - I) = -(I-S)^H\left[I \,\middle|\, -V^H\right] = -N_k^H, \\ &\left[0 \,\middle|\, E^H\right]Q^{(k)H} = E^H\left[0 \,\middle|\, I\right] + E^H V(I-S)^H\left[I \,\middle|\, -V^H\right] = \left[0 \,\middle|\, E^H\right] + E^H V N_k^H,\end{aligned}$$

$$\begin{aligned}H^{(k)} &= -GQ^{(k)H} = N_k F N_k^H + \begin{bmatrix} 0 \\ E \end{bmatrix} N_k^H + N_k\left[0 \,\middle|\, E^H\right]Q^{(k)H} \\ &= N_k(F + E^H V)N_k^H + \begin{bmatrix} 0 \\ E \end{bmatrix} N_k^H + N_k\left[0 \,\middle|\, E^H\right],\end{aligned} \tag{3.23}$$

giving (3.17). The second to last expression for $H^{(k)}$, with $\|N_k\|_2 \leq 2$, gives (3.19). $\square$

The very simple and strong bound (3.19) on the norm of Hermitian $H^{(k)}$, in terms of E_k and F_k in (3.2) and (3.4), is particularly pleasing, and so (3.15) is a very satisfactory result for any reliable implementation of the Hermitian Lanczos process.

We see from (3.18) that N_k is obtained from N_{k-1} by adding a column and a row.

We used MATLAB to compute the eigenvalues of $\left[\begin{smallmatrix} T_k & 0 \\ 0 & A \end{smallmatrix}\right]$ and compare them with those of the matrix on the right-hand side of (3.15). In all our tests on problems with full matrices having n from 30 to 300, and k from 20 to 400, we found the absolute difference of every computed eigenvalue to be less than $nk^{\frac{1}{2}}\epsilon\|A\|_2$ in magnitude, usually significantly so. We chose this comparison since (3.19) with the bounds in section 3.2 gives a bound on $\|H^{(k)}\|_F$ of about $12nk\epsilon\|A\|_2$ for all but very small k.

It would be possible, and perhaps more natural, to rewrite (3.15) and (3.16) as

$$\begin{aligned}&\widetilde{Q}^{(k)H}\left(\begin{bmatrix} A & 0 \\ 0 & T_k \end{bmatrix} + \widetilde{H}^{(k)}\right)\widetilde{Q}^{(k)} = \left[\begin{array}{c|c} T_k & e_k\beta_{k+1}v_{k+1}^H \\ \hline v_{k+1}\beta_{k+1}e_k^T & A_k \end{array}\right], \\ &\widetilde{Q}^{(k)} \triangleq \left[\widetilde{Q}_1^{(k)} \,\middle|\, \widetilde{Q}_2^{(k)}\right] \triangleq \left[\begin{array}{c|c} V_k(I-S_k) & I - V_k(I-S_k)V_k^H \\ S_k & (I-S_k)V_k^H \end{array}\right], \quad \widetilde{Q}^{(k)H} = \{\widetilde{Q}^{(k)}\}^{-1}.\end{aligned}$$

But the analysis is identical, and (possibly because of familiarity) we find (3.15) and (3.16) easier to work with. Also the increasing dimensions of $\left[\begin{smallmatrix} T_k & 0 \\ 0 & A \end{smallmatrix}\right]$ in (3.15) might initially seem disturbing, but they are a necessary result of the analysis.

Note that $Q_1^{(k)}$ in (3.16) is $(n+k)\times k$, while T_k is $k \times k$. The columns of $Q_1^{(k)}$ will be seen to be k orthonormal Lanczos vectors for a strange matrix. The theoretical orthogonal similarity transformation in (3.15) has the form of k rounding-error-free steps of the Householder tridiagonalization of $\left[\begin{smallmatrix} T_k & 0 \\ 0 & A \end{smallmatrix}\right] + H^{(k)}$; see, for example, [8, section 8.3.1]. We now state it as a rounding-error-free Lanczos process of the form of (1.1).

COROLLARY 3.2. *After k finite precision steps of a Lanczos algorithm with $A = A^H$ satisfying* (3.2) *and* (3.4)*, and leading to V_k, v_{k+1}, and $T_{k+1,k}$ in section* 3.1*, with S_k, s_{k+1}, $Q^{(k)}$, and $H^{(k)}$ defined in Theorem* 3.1 *(see also* (3.8)*) we have, along with* (3.16)*,*

$$(3.24)\quad \left(\begin{bmatrix} T_k & 0 \\ 0 & A \end{bmatrix} + H^{(k)}\right) Q_1^{(k)} = \left[\; Q_1^{(k)} \;\middle|\; q_{k+1} \;\right] T_{k+1,k} = Q_1^{(k)} T_k + q_{k+1}\beta_{k+1} e_k^T,$$

$$(3.25)\quad \left[Q_1^{(k)} \;\middle|\; q_{k+1}\right] \triangleq \left[\begin{array}{c|c} S_k & s_{k+1} \\ V_k(I-S_k) & v_{k+1} - V_k s_{k+1} \end{array}\right], \quad [Q_1^{(k)}|q_{k+1}]^H [Q_1^{(k)}|q_{k+1}] = I_{k+1},$$

where q_{k+1} is also equal to the $(k+1)$st column of $Q_1^{(k+1)}$ with its $(k+1)$st element (a zero) removed.

Proof. This follows immediately from $Q^{(k)}$ times (3.15), with (3.16) and

$$(3.26)\qquad Q_2^{(k)} v_{k+1} = \begin{bmatrix} (I-S_k)V_k^H v_{k+1} \\ v_{k+1} - V_k(I-S_k)V_k^H v_{k+1} \end{bmatrix} = \begin{bmatrix} s_{k+1} \\ v_{k+1} - V_k s_{k+1} \end{bmatrix} = q_{k+1},$$

$$(3.27)\qquad Q_1^{(k+1)} e_{k+1} = \begin{bmatrix} S_{k+1} \\ V_{k+1}(I_{k+1} - S_{k+1}) \end{bmatrix} e_{k+1} = \begin{bmatrix} s_{k+1} \\ 0 \\ v_{k+1} - V_k s_{k+1} \end{bmatrix},$$

since from (3.8) $s_{k+1} = (I - S_k)V_k^H v_{k+1}$. ∎

The whole of (3.24) shows how $T_{k+1,k}$ and $[Q_1^{(k)}, q_{k+1}]$ develop, while the first k rows, equivalent to (3.9), show how the loss of orthogonality, S_{k+1}, develops.

Equation (3.24) shows that the columns of $[Q_1^{(k)}, q_{k+1}]$ are the *exact* Lanczos vectors, and $T_{k+1,k}$ the *exact* matrix, for k steps of the Lanczos process with $\left[\begin{smallmatrix} T_k & 0 \\ 0 & A \end{smallmatrix}\right] + H^{(k)}$ and $\left[\begin{smallmatrix} 0 \\ v_1 \end{smallmatrix}\right]$. So, quite unexpectedly, we do have a *backward-like* rounding error result in matrix form for this version of the Lanczos algorithm. But this result has T_k appearing on both sides of the Lanczos equation (3.24), a truly novel occurrence.

Note that E_k and F_k in (3.2) and (3.4) create all the other error terms. E_k contributes to F_k in (3.4), but it is F_k which determines all the loss of orthogonality. Most importantly, $H^{(k)} = 0$ in (3.15) if and only if we have an ideal, error-free Lanczos process with exact orthogonality, showing that this analysis is both complete and tight (all necessary terms are included, and there are no unnecessary terms).

COROLLARY 3.3. *For a finite precision Lanczos algorithm of the form referred to in Theorem* 3.1 *and Corollary* 3.2*,*

(3.28)

$$E_k = 0 \;\&\; \{\text{local orthogonality } u_{i,i+1} \triangleq v_i^H v_{i+1} = 0,\ i = 1, \dots, k,\} \Rightarrow F_k = 0,$$

(3.29)

$$F_k = 0 \Leftrightarrow U_{k+1} = 0 \Leftrightarrow S_{k+1} = 0 \Leftrightarrow V_{k+1}^H V_{k+1} = I_{k+1},$$

(3.30)
$$E_k = 0 \;\&\; F_k = 0 \Leftrightarrow H^{(k)} = 0 \quad \text{in (3.15)},$$

(3.31)
$$H^{(k)} = 0 \Leftrightarrow \text{this is an error-free Lanczos process.}$$

Proof. Equation (3.28) follows from (3.4). In (3.4), $U_{k+1} = 0$ shows that $F_k = 0$, while $F_k = 0$ and the facts that U_{k+1} is SUT and $\beta_2 \cdots \beta_{k+1} \neq 0$ show that $U_{k+1} = 0$, proving the first implication in (3.29). The remaining implications in (3.29) follow from (2.1) and (2.2). Then for (3.30), (3.17) shows that $E_k = 0 \;\&\; F_k = 0 \Rightarrow H^{(k)} = 0$, while from (3.23) $H^{(k)} = -GQ^{(k)H}$, so $H^{(k)} = 0 \Leftrightarrow G = 0$. But (3.21) and (3.22) show that $G = 0 \Rightarrow E_k = 0 \;\&\; F_k = 0$, completing (3.30).

From these results we see that if we have an ideal Lanczos process (1.1), then $E_k = 0$ and $U_{k+1} = 0$ and so $F_k = 0$, and then $H^{(k)} = 0$. Finally if $H^{(k)} = 0$, then $E_k = 0$ and $F_k = 0$, so $S_{k+1} = 0$, $V_{k+1}^H V_{k+1} = I_{k+1}$, and (3.24) and (3.25) give

$$\begin{bmatrix} T_k & 0 \\ 0 & A \end{bmatrix} \begin{bmatrix} 0 \\ V_k \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ V_k & v_{k+1} \end{bmatrix} T_{k+1,k}, \quad V_{k+1}^H V_{k+1} = I_{k+1}.$$

Thus T_k in the leftmost matrix has no effect, and the nontrivial equations correspond to an ideal error-free Lanczos process (1.1), proving (3.31). ∎

In [10] (see also, for example, [17, section 3.9]) it was shown that the computed T_k is equal to that generated by an exact Lanczos process applied to a matrix of larger dimension than A, each of whose eigenvalues is fairly close to an eigenvalue of A. However, the matrix was not simply defined and the bounds were weak, being $O(\epsilon^{\frac{1}{2}})\|A\|_2$ or even $O(\epsilon^{\frac{1}{4}})\|A\|_2$. In Corollary 3.2 the larger-dimensioned matrix is clearly defined and the bounds are $O(\epsilon)\|A\|_2$, but usually a few of its eigenvalues are not close to eigenvalues of A. Because Corollary 3.2 shows a strange augmented form of stability of the Lanczos process, it follows a path initiated by Greenbaum [9, 10]. It also creates a link to the work of Greenbaum, Strakoš, and coworkers who developed many results on the Lanczos process and CG via an analysis based on the fundamental relationship with the theory of orthogonal polynomials and Gauss quadrature of the Riemann–Stieltjes integral; see, for example, [17, 18, 19] and their many references. For example, a property of the Lanczos process is how it can create essentially repeated eigenvalues in T_k corresponding to single eigenvalues of A, and this property is handled quite beautifully with the Gauss quadrature approach. Here is an alternative explanation. It was shown in [20], [23, Theorem 3.1], that any converged eigenvalue of T_k must be within $O(\epsilon)\|A\|_2$ of an eigenvalue of A. With this, (3.24) shows directly from matrix properties that any converged eigenvalue of T_k is then almost a multiple eigenvalue of $\left[\begin{smallmatrix} T_j & 0 \\ 0 & A \end{smallmatrix}\right] + H^{(j)}$ for all $j \geq k$, and so in all probability will eventually appear again for some $j > k$ as yet another eigenvalue of T_j on the right-hand side of (3.24).

Corollary 3.2 immediately leads to the following observation.

COROLLARY 3.4. *After k finite precision steps of a Lanczos algorithm with $A = A^H$ satisfying* (3.2) *and* (3.4), *and leading to V_k, v_{k+1}, and $T_{k+1,k}$ in section* 3.1, *with S_k, s_{k+1}, $Q^{(k)}$, and $H^{(k)}$ defined in Theorem* 3.1, *the columns of $Q_1^{(k)}$ form an orthonormal basis for the kth Krylov subspace generated by the Hermitian matrix $\left[\begin{smallmatrix} T_k & 0 \\ 0 & A \end{smallmatrix}\right] + H^{(k)}$ with the initial vector $\left[\begin{smallmatrix} 0 \\ v_1 \end{smallmatrix}\right]$.*

Proof. This follows immediately from the unreduced tridiagonal form of $T_{k+1,k}$ in (3.24), and $Q_1^{(k)H} Q_1^{(k)} = I$ in (3.25). ∎

3.4. Comments. The work in [24, section 3] suggested the possibility of a theorem like Theorem 3.1, which was arrived as follows. First a theorem like Corollary 3.2 was found, but instead of $H^{(k)}$ there was an error term at the end. From this was found something like the leading $k \times k$ block of (3.15), but again with the error term not in backward form. This allowed Corollary 3.2 to be found, and eventually the main Theorem 3.1 was derived. It was later proved as shown here.

The Golub–Kahan bidiagonalization [7] can be written as a Hermitian Lanczos process, and so the results here can probably be easily altered to handle that case.

Using the rounding error properties of complex computations as described in [14, section 3.6], it can be shown that (3.2) and (3.4) hold for a good implementation of the skew-Hermitian Lanczos process with similar bounds on E_k and F_k, but with real skew-symmetric T_k. It thus seems reasonable to think that a version of Theorem 3.1 will hold for that case too, perhaps with H_k skew-Hermitian.

It might be possible to generalize Theorem 3.1 to handle the unsymmetric Lanczos process [15, p. 266 et seq.] (see also [36, pp. 388–394]) by using the biorthogonal equivalent of Theorem 2.1 that was described in [24, Theorem 7.1]. These ideas could also be considered for other orthogonalization or biorthogonalization algorithms.

Acknowledgments. In particular I want to thank Ivo Panayotov for his suggestions, discussions, and computations regarding this paper. I also thank Anne Greenbaum, Beresford Parlett, Zdenek Strakoš, and two referees for their very useful feedback.

REFERENCES

[1] J. L. Barlow, N. Bosner, and Z. Drmač, *A new stable bidiagonal reduction algorithm*, Linear Algebra Appl., 397 (2005), pp. 35–84.

[2] Å, Björck, *Numerical Methods for Least Squares Problems*, SIAM, Philadelphia, 1996.

[3] Å. Björck and C. C. Paige, *Loss and recapture of orthogonality in the modified Gram–Schmidt algorithm,* SIAM J. Matrix Anal. Appl., 13 (1992), pp. 176–190.

[4] Å. Björck and C. C. Paige, *Solution of augmented linear systems using orthogonal factorizations,* BIT, 34 (1994), pp. 1–24.

[5] J. Donovan and F. Sullivan, *Top* 10 *Algorithms of the Century*, Comput. Sci. Eng., 2 (2000), pp. 22–23.

[6] W. Givens, *Numerical Computation of the Characteristic Values of a Real Symmetric Matrix,* Oak Ridge National Laboratory, Tech. report ORNL-1574, Oak Ridge, TN, 1954.

[7] G. H. Golub and W. Kahan, *Calculating the singular values and pseudo-inverse of a matrix*, SIAM J. Numer. Anal., 2 (1965), pp. 205–224.

[8] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed., The Johns Hopkins University Press, Baltimore, 1996.

[9] A. Greenbaum, *Convergence Properties of the Conjugate Gradient Algorithm in Exact and Finite Precision Arithmetic*, Ph.D. thesis, University of California, Berkeley, CA, 1981.

[10] A. Greenbaum, *Behavior of slightly perturbed Lanczos and conjugate-gradient recurrences*, Linear Algebra Appl., 113 (1989), pp. 7–63.

[11] A. Greenbaum and Z. Strakoš, *Predicting the behavior of finite precision Lanczos and conjugate gradient computations*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 121–137.

[12] M. H. Gutknecht and Z. Strakoš, *Accuracy of two three-term and three two-term recurrences for Krylov space solvers*, SIAM J. Matrix Anal. Appl., 22 (2000), pp. 213–229.

[13] M. R. Hestenes and E. Stiefel, *Methods of conjugate gradients for solving linear systems*, J. Research Nat. Bur. Standards, 49 (1952), pp. 409–436.

[14] N. J. Higham, *Accuracy and Stability of Numerical Algorithms*, 2nd ed., SIAM, Philadelphia, 2002.

[15] C. Lanczos, *An iteration method for the solution of the eigenvalue problem of linear differential and integral operators*, J. Research Nat. Bur. Standards, 45 (1950), pp. 255–282.

[16] C. Lanczos, *Solution of systems of linear equations by minimized-iterations*, J. Research Nat. Bur. Standards, 49 (1952), pp. 33–53.

[17] G. Meurant, *The Lanczos and Conjugate Gradient Algorithms: From Theory to Finite Precision Computations*, Software Environments Tools 19, SIAM, Philadelphia, 2006.

[18] G. Meurant and Z. Strakoš, *The Lanczos and conjugate gradient algorithms in finite precision arithmetic*, Acta Numer., 15 (2006), pp. 471–542.

[19] D. P. O'Leary, Z. Strakoš, and P. Tichý, *On sensitivity of Gauss-Christoffel quadrature*, Numer. Math., 107 (2007), pp. 147–174.

[20] C. C. Paige, *The Computation of Eigenvalues and Eigenvectors of Very Large Sparse Matrices*, Ph.D. thesis, London University, London, UK, 1971.

[21] C. C. Paige, *Computational variants of the Lanczos method for the eigenproblem*, J. Inst. Math. Appl., 10 (1972), pp. 373–381.

[22] C. C. Paige, *Error analysis of the Lanczos algorithm for tridiagonalizing a symmetric matrix*, J. Inst. Math. Appl., 18 (1976), pp. 341–349.

[23] C. C. Paige, *Accuracy and effectiveness of the Lanczos algorithm for the symmetric eigenproblem*, Linear Algebra Appl., 34 (1980), pp. 235–258.

[24] C. C. Paige, *A useful form of unitary matrix obtained from any sequence of unit* 2*-norm* n*-vectors*, SIAM J. Matrix Anal. Appl., 31 (2009), pp. 565–583.

[25] 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.

[26] C. C. Paige and M. A. Saunders, *Solution of sparse indefinite systems of linear equations*, SIAM J. Numer. Anal., 12 (1975), pp. 617–629.

[27] C. C. Paige and M. A. Saunders, *LSQR: An algorithm for sparse linear equations and sparse least squares*, ACM Trans. Math. Software, 8 (1982), pp. 43–71.

[28] C. C. Paige and M. A. Saunders, *ALGORITHM* 583: *LSQR: Sparse linear equations and least squares problems*, ACM Trans. Math. Software, 8 (1982), pp. 195–209.

[29] C. C. Paige and Z. Strakoš, *Scaled total least squares fundamentals*, Numer. Math., 91 (2002), pp. 117–146.

[30] B. N. Parlett, *The Symmetric Eigenvalue Problem*, Classics in Appl. Math. 20, SIAM, Philadelphia, 1998.

[31] J. K. Reid, *On the method of conjugate gradients for the solution of large sparse linear equations*, in Large Sparse Sets of Linear Equations, J. K. Reid, ed., Academic Press, London, 1971, pp. 231–254.

[32] Y. Saad and M. H. Schultz, *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 856–869.

[33] C. Sheffield, *private communication with Gene Golub*, circa 1967.

[34] Z. Strakoš, *On the real convergence rate of the conjugate gradient method*, Linear Algebra Appl., 154/156 (1991), pp. 535–549.

[35] Z. Strakoš, *Convergence and numerical behavior of the Krylov space methods*, in Proceedings of the NATO ASI Institute Algorithms for Large Sparse Linear Algebraic Systems: The State of the Art and Applications in Science and Engineering, G. Winter Althaus and E. Spedicato, eds., Kluwer Academic Publishers, Dordrecht, 1998, pp. 175–197.

[36] J. H. Wilkinson, *The Algebraic Eigenvalue Problem*, Clarendon Press, Oxford University Press, New York, 1988.

[37] W. Wülling, *The stabilization of weights in the Lanczos and conjugate gradient method*, BIT, 45 (2005), pp. 395–414.

[38] W. Wülling, *On stabilization and convergence of clustered Ritz values in the Lanczos method*, SIAM J. Matrix Anal. Appl., 27 (2005), pp. 891–908.

[39] J.-P. M. Zemke, *Krylov Subspace Methods in Finite Precision: A Unified Approach*, Dissertation, Institut für Informatik III, Technische Universität Hamburg-Harburg, Hamburg, Germany, 2003.

[40] J.-P. M. Zemke, *Hessenberg eigenvalue-eigenmatrix relations*, Linear Algebra Appl., 414 (2006), pp. 589–606.