SIAM J. MATRIX ANAL. APPL.
Vol. 36, No. 2, pp. 793–819

ACCURACY OF THE *s*-STEP LANCZOS METHOD FOR THE SYMMETRIC EIGENPROBLEM IN FINITE PRECISION*

ERIN CARSON† AND JAMES W. DEMMEL†

Abstract. The *s*-step Lanczos method can achieve an $O(s)$ reduction in data movement over the classical Lanczos method for a fixed number of iterations, allowing the potential for significant speedups on modern computers. However, although the *s*-step Lanczos method is equivalent to the classical Lanczos method in exact arithmetic, it can behave quite differently in finite precision. Increased roundoff errors can manifest as a loss of accuracy or deterioration of convergence relative to the classical method, reducing the potential performance benefits of the *s*-step approach. In this paper, we present for the first time a complete rounding error analysis of the *s*-step Lanczos method. Our methodology is analogous to Paige's rounding error analysis for classical Lanczos [*IMA J. Appl. Math.*, 18 (1976), pp. 341–349]. Our analysis gives upper bounds on the loss of normality of and orthogonality between the computed Lanczos vectors, as well as a recurrence for the loss of orthogonality. We further demonstrate that bounds on accuracy for the finite precision Lanczos method given by Paige [*Linear Algebra Appl.*, 34 (1980), pp. 235–258] can be extended to the *s*-step Lanczos case assuming a bound on the maximum condition number of the precomputed *s*-step Krylov bases. Our results confirm that the conditioning of the precomputed Krylov bases plays a large role in determining finite precision behavior. In particular, if one can enforce that the condition numbers of the precomputed *s*-step Krylov bases are not too large in any iteration, then the finite precision behavior of the *s*-step Lanczos method will be similar to that of classical Lanczos.

Key words. Krylov subspace methods, error analysis, finite precision, roundoff error, Lanczos, avoiding communication

AMS subject classifications. 65G50, 65F10, 65F15, 65N15, 65N12

DOI. 10.1137/140990735

1. Introduction.

Given a symmetric matrix $A \in \mathbb{R}^{n\times n}$ and a starting vector $v_1 \in \mathbb{R}^n$ with unit 2-norm, m steps of the Lanczos method [28] theoretically produce the orthonormal matrix $V_m = [v_1, \ldots, v_m]$ and the symmetric tridiagonal matrix $T_m \in \mathbb{R}^{m\times m}$ such that

$$AV_m = V_m T_m + \beta_{m+1} v_{m+1} e_m^T. \tag{1.1}$$

When $m = n$, if T_n exists (i.e., no breakdown occurs), then the eigenvalues of T_n are the eigenvalues of A. In practice, some of the eigenvalues of T_m are good approximations to the eigenvalues of A when $m \ll n$, which makes the Lanczos method attractive as an iterative procedure. Many Krylov subspace methods (KSMs), including those for solving linear systems and least squares problems, are based on the Lanczos method. Such Lanczos-based methods are the core components in many applications.

*Received by the editors October 8, 2014; accepted for publication (in revised form) by J. L. Barlow March 20, 2015; published electronically June 18, 2015. The research of the authors was supported by the U.S. Department of Energy Office of Science, Office of Advanced Scientific Computing Research, Applied Mathematics program under awards DE-SC0004938, DE-SC0003959, and DE-SC0010200, by the U.S. Department of Energy Office of Science, Office of Advanced Scientific Computing Research, X-Stack program under awards DE-SC0005136, DE-SC0008699, DE-SC0008700, and AC02-05CH11231, and by DARPA award HR0011-12-2-0016, with contributions from Intel, Oracle, and MathWorks.

http://www.siam.org/journals/simax/36-2/99073.html

†EECS Computer Science Division, University of California, Berkeley, Berkeley, CA 94720-1776 (ecc22@eecs.berkeley.edu, demmel@cs.berkeley.edu).

Classical implementations of Krylov methods, the Lanczos method included, require one or more sparse matrix-vector multiplications (SpMVs) and one or more inner product operations in each iteration. These computational kernels are both communication-bound on modern computer architectures. To perform an SpMV, each processor must communicate entries of the source vector it owns to other processors in the parallel algorithm, and in the sequential algorithm the matrix A must be read from slow memory (when it is too large to fit in cache, the most interesting case). Inner products involve a global reduction (see [42, section 11.4]) in the parallel algorithm and a number of reads and writes to slow memory in the sequential algorithm (depending on the size of the vectors and the size of the fast memory).

Thus, many efforts have focused on communication-avoiding Krylov subspace methods (CA-KSMs), or s-step Krylov methods, which can perform s iterations with a factor of $O(s)$ less communication than classical KSMs; see, e.g., [7, 8, 10, 12, 13, 15, 22, 23, 45, 47]. In practice, this can translate into significant speedups for many problems [32, 50]. In this paper, we will use the terminology "s-step methods," which was introduced in [9]. The reader should note this use of the term differs from other works, e.g., [11, 25] and [17, section 9.2.7], in which the term "s-step methods" is used to refer to a type of restarted Lanczos procedure.

Equally important to the performance of each iteration is the convergence rate of the method, i.e., the total number of iterations required until the desired convergence criterion is met. Although theoretically the Lanczos process described by (1.1) produces an orthogonal basis and a tridiagonal matrix similar to A after n steps, these properties need not hold in finite precision.

Although s-step Krylov methods are mathematically equivalent to their classical counterparts in exact arithmetic, their finite precision behavior may differ significantly. It has been observed that the behavior of s-step Krylov methods deviates further from that of the classical method as s increases and the severity of this deviation is heavily influenced by the polynomials used for the s-step Krylov bases (see, e.g., [1, 7, 23, 24]).

The most important work in the finite precision analysis of classical Lanczos is a series of papers published by Paige [33, 34, 35, 36]. Paige's analysis succinctly describes how rounding errors propagate through the algorithm to impede orthogonality. These results were developed to give theorems which link the loss of orthogonality to convergence of the computed eigenvalues [36]. Until now, no analogous theory existed for the s-step Lanczos method.

In this paper, we present for the first time a complete rounding error analysis of the s-step Lanczos method. We provide upper bounds on the normality of and orthogonality between the computed Lanczos vectors as well as a recurrence for the loss of orthogonality. We use this analysis to extend the results of Paige for classical Lanczos to the s-step Lanczos method. Our analysis here of s-step Lanczos very closely follows Paige's rounding error analysis for classical Lanczos [35], and the theorems on accuracy and convergence of eigenvalues presented here follow from [36]. The derived bounds are very similar to those of Paige for classical Lanczos, but with the addition of an amplification term which depends on the maximum condition number of the Krylov bases computed at the start of each block of s steps. We show here that, based on restrictions on the size of this condition number, the same theorems of Paige apply to the s-step case.

Our results confirm that the conditioning of the Krylov bases plays a large role in determining finite precision behavior. Our analysis shows that if one can guarantee that the condition numbers of the computed s-step Krylov bases are not too large throughout the iterations, the accuracy and convergence of eigenvalues in the s-step

Lanczos method will be similar to those produced by classical Lanczos. This indicates that under certain restrictions, the s-step Lanczos method may be suitable for many practical problems.

The remainder of this paper is outlined as follows. In section 2, we present related work in s-step Krylov methods and the analysis of finite precision Lanczos. In section 3, we review a variant of the Lanczos method and derive the corresponding s-step Lanczos method. We also motivate the use of s-step methods in practice, with a discussion of communication-avoiding kernels and recent speedup results. In section 4, we first state our main result in Theorem 4.2 and comment on its interpretation; the rest of the section is devoted to its proof. Sections 5 and 6 use the results of Paige [36] to give results on the accuracy and rate of convergence of the computed eigenvalues, respectively. Section 7 concludes with a discussion of future work.

2. Related work. We briefly review related work in s-step Krylov methods as well as work related to the analysis of classical Krylov methods in finite precision.

2.1. s-step Krylov subspace methods. The term s-step Krylov method, first used by Chronopoulos and Gear [9], describes variants of Krylov methods where the iterations are performed in blocks of s at a time. Since the Krylov subspaces required to perform s iterations of updates are known, bases for these subspaces are computed at the start of each block of s steps, inner products between basis vectors are computed with one block inner product, and then s iterations are performed by updating the coordinates in the generated Krylov bases. (See section 3 for details.) Many variations have been derived over the past few decades with various motivations, namely, increasing parallelism (e.g., [9, 47, 48]) and avoiding data movement, both between levels of the memory hierarchy in sequential methods and between processors in parallel methods. A thorough treatment of related work can be found in [23].

Many empirical studies of s-step Krylov methods found that convergence often deteriorated using $s > 5$ due to the inherent instability of the monomial basis. This motivated research into the use of better-conditioned bases (e.g., Newton or Chebyshev polynomials) for the Krylov subspace, which allowed convergence for higher s values (see, e.g., [1, 22, 24, 40]).

The term "communication-avoiding Krylov methods" refers to s-step Krylov methods and implementations which aim to improve performance by asymptotically decreasing communication costs, possibly both in computing inner products and computing the s-step bases, for both sequential and parallel algorithms; see [13, 23]. Hoemmen and co-authors [23, 32] derived communication-avoiding variants of Lanczos, Arnoldi, conjugate gradient (CG) and the generalized minimum residual method (GMRES). Details of nonsymmetric Lanczos-based CA-KSMs, including communication-avoiding versions of biconjugate gradient (BICG) and stabilized biconjugate gradient (BICGSTAB) can be found in [7]. Although potential performance improvement is our primary motivation for studying these methods, we use the general term "s-step methods" here as our error analysis is independent of performance.

Many efforts have been devoted specifically to the s-step Lanczos method. The first s-step Lanczos methods known in the literature are due to Kim and Chronopoulos, who derived a three-term symmetric s-step Lanczos method [26] as well as a three-term nonsymmetric s-step Lanczos method [27]. Hoemmen derived a three-term communication-avoiding Lanczos method, CA-Lanczos [23]. A two-term communication-avoiding nonsymmetric Lanczos method (called CA-BIOC, based on the "BIOC" version of nonsymmetric Lanczos of Gutknecht [21]) can be found in [2]. Section 3 of the present work includes the derivation of a new version of the s-step Lanczos

method, equivalent in exact arithmetic to the variant used by Paige [35]. It uses a two-term recurrence like BIOC but is restricted to the symmetric case and uses a different starting vector.

For s-step KSMs that solve linear systems, increased roundoff error can decrease the maximum attainable accuracy of the solution, resulting in a less accurate solution than that found by the classical method. A quantitative analysis of roundoff error in CA-CG and CA-BICG can be found in [6]. Based on the work of [46] for classical KSMs, we have also explored implicit residual replacement for CA-CG and CA-BICG as a method to limit the deviation of true and updated residuals when high accuracy is required [6].

2.2. Error analysis of the Lanczos method. Lanczos and others recognized early on that rounding errors could cause the Lanczos method to deviate from its ideal theoretical behavior. Since then, various efforts have been devoted to analyzing, explaining, and improving the finite precision Lanczos method.

Widely considered to be the most significant development is the series of papers by Paige discussed in section 1. Another important result is due to Greenbaum, who performed a backward-like error analysis which showed that finite precision Lanczos and CG behave very similarly to the exact algorithms applied to any of a certain class of larger matrices [18]. Further explanation and examples are due to Greenbaum and Strakoš [19]. Paige has shown a similar type of augmented stability for the Lanczos process [37], and these results have recently been extended to the nonsymmetric case [38]. There are many other analyses of the behavior of various KSMs in finite precision, including some more recent results due to Wülling [51] and Zemke [52]; for a thorough overview of the literature, see [29, 30].

A number of strategies for maintaining orthogonality among the Lanczos vectors were inspired by the analysis of Paige, including selective reorthogonalization [39] and partial reorthogonalization [43]. Recently, Gustafsson, Demmel, and Holmgren have extended such reorthogonalization strategies for classical Lanczos to the s-step case [20].

3. The *s*-step Lanczos method. In our analysis, we use the same variant of Lanczos used by Paige in [35] to allow easy comparison of results. Note that for simplicity, we assume no breakdown occurs, i.e., $\beta_{i+1} \neq 0$ for $i < n$, and thus breakdown conditions are not discussed here. We now give a derivation of s-step Lanczos, obtained from classical Lanczos in Algorithm 1.

After k blocks of s steps, consider iteration $i = sk+1$, where $k \in \mathbb{N}$ and $0 < s \in \mathbb{N}$. Using

$$v_{sk+1} \in \mathcal{K}_1(A, v_{sk+1}) \quad \text{and} \quad u_{sk+1} \in \mathcal{K}_1(A, u_{sk+1}),$$

Algorithm 1. The classical Lanczos method.

Require: n-by-n real symmetric matrix A and length-n vector v_1 such that $\|v_1\|_2 = 1$

1: $u_1 = Av_1$
2: **for** $i = 1, 2, \ldots$ until convergence **do**
3: $\quad \alpha_i = v_i^T u_i$
4: $\quad w_i = u_i - \alpha_i v_i$
5: $\quad \beta_{i+1} = \|w_i\|_2$
6: $\quad v_{i+1} = w_i / \beta_{i+1}$
7: $\quad u_{i+1} = Av_{i+1} - \beta_{i+1} v_i$
8: **end for**

it follows by induction on lines 6 and 7 of Algorithm 1 that for $j \in \{1, \ldots, s+1\}$,

$$\begin{aligned} v_{sk+j} &\in \mathcal{K}_s(A, v_{sk+1}) + \mathcal{K}_s(A, u_{sk+1}) \quad \text{and} \\ u_{sk+j} &\in \mathcal{K}_{s+1}(A, v_{sk+1}) + \mathcal{K}_{s+1}(A, u_{sk+1}), \end{aligned} \tag{3.1}$$

where $\mathcal{K}_\ell(A, x) = \text{span}\{x, Ax, \ldots, A^{\ell-1}x\}$ denotes the Krylov subspace of dimension ℓ of matrix A with respect to vector x. Since $\mathcal{K}_j(A, x) \subseteq \mathcal{K}_\ell(A, x)$ for $j \leq \ell$,

$$v_{sk+j}, u_{sk+j} \in \mathcal{K}_{s+1}(A, v_{sk+1}) + \mathcal{K}_{s+1}(A, u_{sk+1}) \quad \text{for} \quad j \in \{1, \ldots, s+1\}.$$

Note that since $u_1 = Av_1$, if $k = 0$, we have

$$v_j, u_j \in \mathcal{K}_{s+2}(A, v_1) \quad \text{for} \quad j \in \{1, \ldots, s+1\}.$$

For $k > 0$, we then define the "basis matrix" $\mathcal{Y}_k = [\mathcal{V}_k, \mathcal{U}_k]$, where $\mathcal{V}_k$ and $\mathcal{U}_k$ are size n-by-$(s+1)$ matrices whose first j columns form bases for $\mathcal{K}_j(A, v_{sk+1})$ and $\mathcal{K}_j(A, u_{sk+1})$, respectively, for $j \in \{1, \ldots, s+1\}$. For $k = 0$, we define $\mathcal{Y}_0$ to be a size n-by-$(s+2)$ matrix whose (ordered) columns form a basis for $\mathcal{K}_{s+2}(A, v_1)$. Then by (3.1), we can represent v_{sk+j} and u_{sk+j}, for $j \in \{1, \ldots, s+1\}$, by their coordinates (denoted with primes) in $\mathcal{Y}_k$, i.e.,

$$v_{sk+j} = \mathcal{Y}_k v'_{k,j} \quad \text{and} \quad u_{sk+j} = \mathcal{Y}_k u'_{k,j}. \tag{3.2}$$

Note that for $k = 0$, the coordinate vectors $v'_{k,j}$ and $u'_{k,j}$ have length $s+2$ with zero elements beyond the jth and $j+1$st, respectively, and for $k > 0$, the coordinate vectors have length $(2s+2)$ with appropriate zero elements. We can write a similar equation for auxiliary vector w_{sk+j}, i.e., $w_{sk+j} = \mathcal{Y}_k w'_{k,j}$ for $j \in \{1, \ldots, s\}$. We also define the Gram matrix $G_k = \mathcal{Y}_k^T \mathcal{Y}_k$, which is size $(s+2)$-by-$(s+2)$ for $k = 0$ and $(2s+2)$-by-$(2s+2)$ for $k > 0$. Using this matrix, the inner products in lines 3 and 5 can be written

$$\alpha_{sk+j} = v_{sk+j}^T u_{sk+j} = v'^T_{k,j} \mathcal{Y}_k^T \mathcal{Y}_k u'_{k,j} = v'^T_{k,j} G_k u'_{k,j} \quad \text{and} \tag{3.3}$$

$$\beta_{sk+j+1} = (w_{sk+j}^T w_{sk+j})^{1/2} = (w'^T_{k,j} \mathcal{Y}_k^T \mathcal{Y}_k w'_{k,j})^{1/2} = (w'^T_{k,j} G_k w'_{k,j})^{1/2}. \tag{3.4}$$

We assume that the bases are generated via polynomial recurrences represented by matrix $\mathcal{B}_k$, which is in general upper Hessenberg but often tridiagonal in practice. The recurrence can thus be written in matrix form as

$$A\underline{\mathcal{Y}}_k = \mathcal{Y}_k \mathcal{B}_k. \tag{3.5}$$

For $k = 0$, $\mathcal{B}_k$ is size $(s+2)$-by-$(s+2)$ and $\underline{\mathcal{Y}}_0$ is obtained by replacing the last column of $\mathcal{Y}_0$ by a zero column, i.e., $\underline{\mathcal{Y}}_0 = \left[\mathcal{Y}_0[I_{s+1}, 0_{s+1,1}]^T, 0_{n,1}\right]$. For $k > 0$, $\mathcal{B}_k$ is size $(2s+2)$-by-$(2s+2)$ and $\underline{\mathcal{Y}}_k$ is obtained by replacing columns $s+1$ and $2s+2$ of $\mathcal{Y}_k$ by zero columns, i.e., $\underline{\mathcal{Y}}_k = \left[\mathcal{V}_k[I_s, 0_{s,1}]^T, 0_{n,1}, \mathcal{U}_k[I_s, 0_{s,1}]^T, 0_{n,1}\right]$. Note that $\mathcal{B}_k$ has zeros in column $s+2$ when $k = 0$ and zeros in columns $s+1$ and $2s+2$ for $k > 0$. Therefore, using (3.2) and (3.5),

$$Av_{sk+j+1} = \mathcal{Y}_k \mathcal{B}_k v'_{k,j+1} \quad \text{for} \quad j \in \{1, \ldots, s\}. \tag{3.6}$$

Thus, to compute iterations $sk+2$ through $sk+s+1$ in s-step Lanczos, we first generate a basis matrix $\mathcal{Y}_k$ such that (3.5) holds and compute G_k from $\mathcal{Y}_k$. Then updates to the length-n vectors can be performed by updating instead the length-

ALGORITHM 2. THE s-STEP LANCZOS METHOD.

Require: n-by-n real symmetric matrix A and length-n vector v_1 such that $\|v_1\|_2 = 1$

1: $u_1 = Av_1$
2: **for** $k = 0, 1, \ldots$ until convergence **do**
3: Compute $\mathcal{Y}_k$ with change of basis matrix $\mathcal{B}_k$
4: Compute $G_k = \mathcal{Y}_k^T \mathcal{Y}_k$
5: $v'_{k,1} = e_1$
6: **if** $k = 0$ **then**
7: $u'_{0,1} = \mathcal{B}_0 e_1$
8: **else**
9: $u'_{k,1} = e_{s+2}$
10: **end if**
11: **for** $j = 1, 2, \ldots, s$ **do**
12: $\alpha_{sk+j} = v'^T_{k,j} G_k u'_{k,j}$
13: $w'_{k,j} = u'_{k,j} - \alpha_{sk+j} v'_{k,j}$
14: $\beta_{sk+j+1} = (w'^T_{k,j} G_k w'_{k,j})^{1/2}$
15: $v'_{k,j+1} = w'_{k,j} / \beta_{sk+j+1}$
16: $v_{sk+j+1} = \mathcal{Y}_k v'_{k,j+1}$
17: $u'_{k,j+1} = \mathcal{B}_k v'_{k,j+1} - \beta_{sk+j+1} v'_{k,j}$
18: $u_{sk+j+1} = \mathcal{Y}_k u'_{k,j+1}$
19: **end for**
20: **end for**

$(2s + 2)$ coordinates for those vectors in $\mathcal{Y}_k$. Inner products and multiplications with A become smaller operations which can be performed locally, as in (3.3), (3.4), and (3.6). The complete s-step Lanczos algorithm is presented in Algorithm 2. Note that in Algorithm 2, we show the length-n vector updates in each inner iteration (lines 16 and 18) for clarity, although these vectors play no part in the inner loop iteration updates. In practice, the basis change operation (3.2) can be performed on a block of coordinate vectors at the end of each outer loop to recover v_{sk+j} and u_{sk+j}, if needed, for $j \in \{2, \ldots, s + 1\}$.

3.1. Communication-avoiding kernels. The CA-KSMs introduced by Hoemmen, Mohiyuddin, and others (see [23]), as well as the s-step Lanczos method in Algorithm 2 above, are designed to allow use of communication-avoiding kernels which can asymptotically reduce communication cost. The matrix powers kernel optimization fuses together a sequence of s SpMV operations into one kernel invocation. This kernel is used to compute the $(s + 1)$-dimensional Krylov bases $\mathcal{K}_{s+1}(A, v_{sk+1})$ and $\mathcal{K}_{s+1}(A, u_{sk+1})$ in Algorithm 2 . Depending on the nonzero structure of A (more precisely, of $\{A^j\}_{j=1}^s$), this enables communication-avoidance in both serial and parallel implementations, as described in paragraphs below. For an in-depth treatment of the matrix powers kernel implementation, see [14].

Serial. In serial, the matrix powers kernel reorganizes the s SpMVs to maximize reuse of A and the $s+1$ vectors. This means ideally reading A and the starting vector only once and writing the s output vectors spanning the Krylov subspace only once. When the communication cost of reading A dominates that of reading/writing the vectors (a common situation), this results in an s-fold decrease in both latency (the number of messages sent) and bandwidth (the number of words moved).

Parallel. In a parallel implementation, the matrix powers kernel reorganizes the computation in a similar way but with a slightly different goal. In a parallel SpMV

operation, only entries of the vectors need to be communicated. The parallel matrix powers kernel avoids interprocessor synchronization by initially storing some redundant elements of A and the starting vector on different processors and performing redundant computation to compute the s Krylov basis vectors without further synchronization in between SpMVs. Provided the additional bandwidth and latency cost to distribute the starting vector is a lower-order term (equivalently, A^s is *well partitioned*; see [14]), this gives an s-fold savings in latency cost.

Serial and parallel variants of the matrix powers kernel, for both structured and general sparse matrices, are described in [31] and [2], which summarize most of [14] and elaborate on the implementation in [32]. Within [31], we refer the reader to the complexity analysis in Tables 2.3–4, the performance modeling in section 2.6, and the performance results in section 2.10.3 and section 2.11.3, which demonstrate that this optimization leads to speedups in practice.

For example, for a two-dimensional five-point stencil on a $\sqrt{n} \times \sqrt{n}$ mesh with p processors, assuming $s \ll \sqrt{n/p}$, the number of arithmetic operations grows by a factor $1 + 2s\sqrt{p/n}$, the number of messages decreases by a factor of $s/2$, and the number of words moved grows by a factor of $1 + (s/2)\sqrt{p/n}$ [31]. Therefore, since the additional arithmetic operations and additional words moved are lower-order terms, we expect to see a $\Theta(s)$ speedup when latency is the dominant cost. We note that matrix powers kernel performance is sensitive to matrix structure and hardware parameters, making it a good candidate for inclusion in auto-tuning libraries and specializers.

Besides SpMV operations, classical KSMs also must compute inner products in each iteration, which incur a costly global synchronization on parallel computers. For the s-step Lanczos method in Algorithm 2, inner products are computed as a block operation producing the Gram matrix G_k. In parallel, this can lead to an s-fold decrease in latency.

These communication-avoiding variants can lead to speedups in practice. We direct the reader to recent performance results in [50], which demonstrate speedups up to 4.2 for a communication-avoiding BICGSTAB implementation with $s = 4$.

4. The *s*-step Lanczos method in finite precision. Throughout our analysis, we use a standard model of floating point arithmetic where we assume the computations are carried out on a machine with relative precision ϵ (see [16]). The analysis is first order in ϵ, ignoring higher-order terms, which have negligible effect on our results. We also ignore underflow and overflow. Following Paige [35], we use the ϵ symbol to represent the relative precision as well as terms whose absolute values are bounded by the relative precision.

We will model floating point computation using the following standard conventions (see, e.g., [16, section 2.4]): for vectors $u, v \in \mathbb{R}^n$, matrices $A \in \mathbb{R}^{n \times m}$ and $G \in \mathbb{R}^{n \times n}$, and scalar α,

$$\begin{aligned}
fl(u - \alpha v) &= u - \alpha v - \delta w, & |\delta w| &\le (|u| + 2|\alpha v|)\epsilon, \\
fl(v^T u) &= (v + \delta v)^T u, & |\delta v| &\le n\epsilon |v|, \\
fl(Au) &= (A + \delta A)u, & |\delta A| &\le m\epsilon |A|, \\
fl(A^T A) &= A^T A + \delta E, & |\delta E| &\le n\epsilon |A^T||A|, \quad \text{and} \\
fl(u^T (Gv)) &= (u + \delta u)^T (G + \delta G)v, & |\delta u| &\le n\epsilon |u|, |\delta G| \le n\epsilon |G|,
\end{aligned}$$

where $fl()$ represents the evaluation of the given expression in floating point arithmetic and terms with δ denote error terms. We decorate quantities computed in finite

precision arithmetic with hats, e.g., if we are to compute the expression $\alpha = v^T u$ in finite precision we get $\hat{\alpha} = fl(v^T u)$.

We first prove the following lemma which will be useful in our analysis.

LEMMA 4.1. *Assume we have a full rank matrix $Y \in \mathbb{R}^{n\times r}$, where $n \geq r$. Let Y^+ denote the pseudoinverse of Y, i.e., $Y^+ = (Y^T Y)^{-1} Y^T$. Then for any vector $x \in \mathbb{R}^r$,*

$$\| |Y|\,|x| \|_2 \leq \| |Y| \|_2 \|x\|_2 \leq \Gamma \|Yx\|_2,$$

where $\Gamma = \|Y^+\|_2 \| |Y| \|_2 \leq \sqrt{r}\,\|Y^+\|_2 \|Y\|_2$.

Proof. We have

$$\| |Y||x| \|_2 \leq \| |Y| \|_2 \|x\|_2 \leq \| |Y| \|_2 \|Y^+ Yx\|_2 \leq \| |Y| \|_2 \|Y^+\|_2 \|Yx\|_2 = \Gamma \|Yx\|_2. \qquad \square$$

We note that the term Γ can be thought of as a type of condition number for the matrix Y. In the analysis, we will apply the above lemma to the computed "basis matrix" $\hat{\mathcal{Y}}_k$. We assume throughout that the generated bases $\hat{\mathcal{U}}_k$ and $\hat{\mathcal{V}}_k$ are numerically full rank. That is, all singular values of $\hat{\mathcal{U}}_k$ and $\hat{\mathcal{V}}_k$ are greater than $\epsilon n \cdot 2^{\lfloor \log_2 \theta_1 \rfloor}$, where θ_1 is the largest singular value of A. The results of this section are summarized in the following theorem.

THEOREM 4.2. *Assume that Algorithm 2 is implemented in floating point with relative precision ϵ and applied for m steps to the n-by-n real symmetric matrix A with at most N nonzeros per row, starting with vector v_1 with $\|v_1\|_2 = 1$. Let $\sigma \equiv \|A\|_2$, $\theta\sigma = \||A|\|_2$, and $\tau_k\sigma = \||\mathcal{B}_k|\|_2$, where $\mathcal{B}_k$ is defined in (3.5). Let $\Gamma_k = \|\hat{\mathcal{Y}}_k^+\|_2 \| |\hat{\mathcal{Y}}_k| \|_2$, where the superscript "+" denotes the Moore–Penrose pseudoinverse, i.e., $\hat{\mathcal{Y}}_k^+ = (\hat{\mathcal{Y}}_k^T \hat{\mathcal{Y}}_k)^{-1} \hat{\mathcal{Y}}_k^T$, and let*

$$\bar{\Gamma}_k = \max_{\ell \in \{0,\ldots,k\}} \Gamma_\ell \geq 1 \quad \textit{and} \quad \bar{\tau}_k = \max_{\ell \in \{0,\ldots,k\}} \tau_\ell. \tag{4.1}$$

Then for all $i \in \{1,\ldots,m\}$, $\hat{\alpha}_i$, $\hat{\beta}_{i+1}$, and $\hat{v}_{i+1}$ will be computed such that

$$A\hat{V}_m = \hat{V}_m \hat{T}_m + \hat{\beta}_{m+1} \hat{v}_{m+1} e_m^T - \delta\hat{V}_m, \tag{4.2}$$

where

$$\hat{V}_m = [\hat{v}_1, \hat{v}_2, \ldots, \hat{v}_m] \tag{4.3}$$

$$\delta\hat{V}_m = [\delta\hat{v}_1, \delta\hat{v}_2, \ldots, \delta\hat{v}_m] \tag{4.4}$$

$$\hat{T}_m = \begin{bmatrix} \hat{\alpha}_1 & \hat{\beta}_2 & & \\ \hat{\beta}_2 & \ddots & \ddots & \\ & \ddots & \ddots & \hat{\beta}_m \\ & & \hat{\beta}_m & \hat{\alpha}_m \end{bmatrix} \tag{4.5}$$

with

$$\|\delta\hat{v}_i\|_2 \leq \epsilon_1 \sigma, \tag{4.6}$$

$$\hat{\beta}_{i+1} |\hat{v}_i^T \hat{v}_{i+1}| \leq \epsilon_0 \sigma, \tag{4.7}$$

$$|\hat{v}_{i+1}^T \hat{v}_{i+1} - 1| \leq \epsilon_0/2, \quad \textit{and} \tag{4.8}$$

$$\left| \hat{\beta}_{i+1}^2 + \hat{\alpha}_i^2 + \hat{\beta}_i^2 - \|A\hat{v}_i\|_2^2 \right| \leq 2i(3\epsilon_0 + 2\epsilon_1)\sigma^2, \tag{4.9}$$

where

$$\epsilon_0 \equiv 2\epsilon(n+11s+15)\bar{\Gamma}_k^2 \quad \textit{and} \quad \epsilon_1 \equiv \epsilon\big((N+2s+5)\theta+(4s+9)\bar{\tau}_k+(10s+16)\big)\bar{\Gamma}_k. \tag{4.10}$$

Furthermore, if R_m is the strictly upper triangular matrix such that

$$\hat{V}_m^T\hat{V}_m = R_m^T + \mathrm{diag}(\hat{V}_m^T\hat{V}_m) + R_m, \tag{4.11}$$

then

$$\hat{T}_m R_m - R_m\hat{T}_m = \hat{\beta}_{m+1}\hat{V}_m^T\hat{v}_{m+1}e_m^T + \delta R_m, \tag{4.12}$$

where δR_m is upper triangular with elements η such that

$$\begin{aligned} |\eta_{1,1}| &\le \epsilon_0\sigma, \quad \textit{and for } \ell \in \{2,\dots,m\},\\ |\eta_{\ell,\ell}| &\le 2\epsilon_0\sigma,\\ |\eta_{\ell-1,\ell}| &\le (\epsilon_0+2\epsilon_1)\sigma, \quad \textit{and}\\ |\eta_{t,\ell}| &\le 2\epsilon_1\sigma, \ \textit{for } \ t \in \{1,\dots,\ell-2\}. \end{aligned} \tag{4.13}$$

Comments. This generalizes Paige [35] as follows. The bounds in Theorem 4.2 give insight into how orthogonality is lost in the finite precision s-step Lanczos algorithm. Equation (4.6) bounds the error in the columns of the resulting perturbed Lanczos recurrence. How far the Lanczos vectors can deviate from unit 2-norm is given in (4.8), and (4.7) bounds how far adjacent vectors are from being orthogonal. The bound in (4.9) describes how close the columns of $A\hat{V}_m$ and $\hat{T}_m$ are in size. Finally, (4.12) can be thought of as a recurrence for the loss of orthogonality between Lanczos vectors and shows how errors propagate through the iterations.

One thing to notice about the bounds in Theorem 4.2 is that they depend heavily on the term $\bar{\Gamma}_k$, which is a measure of the conditioning of the computed s-step Krylov bases. This indicates that if $\bar{\Gamma}_k$ is controlled in some way to be near constant, i.e., $\bar{\Gamma}_k = O(1)$, the bounds in Theorem 4.2 will be on the same order as Paige's analogous bounds for classical Lanczos [35], and thus we can expect orthogonality to be lost at a similar rate. The bounds also suggest that for the s-step variant to have any use, we must have $\bar{\Gamma}_k = o(\epsilon^{-1/2})$. Otherwise, there can be no expectation of orthogonality.

We have sacrificed some tightness in the bounds in Theorem 4.2 in favor of simplified notation. Particularly, the use of $\bar{\Gamma}_k$ as defined in (4.1) in our bounds is likely to result in a large overestimate of the error. This causes our bounds for the $s = 1$ case to be larger than those of Paige for classical Lanczos. To obtain tighter bounds, in iteration $i \equiv sk + j$, one could instead use, e.g.,

$$\bar{\Gamma}_k \equiv \max\left\{ \frac{\|\,|\hat{\mathcal{Y}}_\ell||\mathcal{B}_\ell||\hat{v}'_{\ell,j'}|\,\|_2}{\|\,|\mathcal{B}_\ell|\,\|_2\|\hat{\mathcal{Y}}_\ell\hat{v}'_{\ell,j'}\|_2}, \max_{x\in\{\hat{w}'_{\ell,j'},\hat{u}'_{\ell,j'},\hat{v}_{\ell,j'},\hat{v}_{\ell,j'+1}\}} \frac{\|\,|\hat{\mathcal{Y}}_\ell||x|\,\|_2}{\|\hat{\mathcal{Y}}_\ell x\|_2} \right\},$$

where the maximum is over $\ell \le k$, $j' \le j$ if $\ell = k$, and $j' \le s$ if $\ell < k$ (see [5, section 5]).

4.1. Proof of Theorem 4.2. The remainder of this section is dedicated to the proof of Theorem 4.2. To simplify the exposition, we have omitted some intermediate steps in the algebra; for the unfamiliar reader, a much longer analysis including the intermediate steps can be found in [4]. Also, in the interest of reducing overbearing subscripts, we use the indexing $i \equiv sk+j$ for quantities produced by both classical and

s-step Lanczos (namely, the elements of the tridiagonal $\hat{T}_m$ and the length-n iteration vectors). We first proceed toward proving (4.8).

In finite precision, the Gram matrix construction in line 4 of Algorithm 2 becomes

$$\hat{G}_k = fl(\hat{\mathcal{Y}}_k^T \hat{\mathcal{Y}}_k) = \hat{\mathcal{Y}}_k^T \hat{\mathcal{Y}}_k + \delta G_k, \quad \text{where} \quad |\delta G_k| \leq \epsilon n |\hat{\mathcal{Y}}_k^T||\hat{\mathcal{Y}}_k|, \tag{4.14}$$

and line 14 of Algorithm 2 becomes $\hat{\beta}_{i+1} = fl(\ fl(\hat{w}_{k,j}'^T \hat{G}_k \hat{w}_{k,j}')^{1/2})$. Let

$$d = fl(\hat{w}_{k,j}'^T \hat{G}_k \hat{w}_{k,j}') = (\hat{w}_{k,j}'^T + \delta \hat{w}_{k,j}'^T)(\hat{G}_k + \delta \hat{G}_{k,w_j})\hat{w}_{k,j}',$$

where

$$|\delta \hat{w}_{k,j}'| \leq \epsilon(2s+2)|\hat{w}_{k,j}'| \quad \text{and} \quad |\delta \hat{G}_{k,w_j}| \leq \epsilon(2s+2)|\hat{G}_k|. \tag{4.15}$$

Remember that in the above equation, we have ignored all terms of second order in ϵ. Now, we let $c = \hat{w}_{k,j}'^T \delta G_k \hat{w}_{k,j}' + \hat{w}_{k,j}'^T \delta \hat{G}_{k,w_j} \hat{w}_{k,j}' + \delta \hat{w}_{k,j}'^T \hat{G}_k \hat{w}_{k,j}'$, where

$$|c| \leq \epsilon(n+4s+4)\Gamma_k^2 \|\hat{\mathcal{Y}}_k \hat{w}_{k,j}'\|_2^2. \tag{4.16}$$

We can then write $d = \|\hat{\mathcal{Y}}_k \hat{w}_{k,j}'\|_2^2(1 + c/\|\hat{\mathcal{Y}}_k \hat{w}_{k,j}'\|_2^2)$ and

$$\hat{\beta}_{i+1} = fl(\sqrt{d}) = \sqrt{d} + \delta\beta_{i+1} = \|\hat{\mathcal{Y}}_k \hat{w}_{k,j}'\|_2 \left(1 + \frac{c}{2\|\hat{\mathcal{Y}}_k \hat{w}_{k,j}'\|_2^2}\right) + \delta\beta_{i+1}, \tag{4.17}$$

where

$$|\delta\beta_{i+1}| \leq \epsilon\sqrt{d} = \epsilon\|\hat{\mathcal{Y}}_k \hat{w}_{k,j}'\|_2. \tag{4.18}$$

The coordinate vector $\hat{v}_{k,j+1}'$ is computed as

$$\hat{v}_{k,j+1}' = fl(\hat{w}_{k,j}'/\hat{\beta}_{i+1}) = (\hat{w}_{k,j}' + \delta\tilde{w}_{k,j}')/\hat{\beta}_{i+1}, \tag{4.19}$$

where

$$|\delta\tilde{w}_{k,j}'| \leq \epsilon|\hat{w}_{k,j}'|. \tag{4.20}$$

The corresponding Lanczos vector $\hat{v}_{i+1}$ (as well as $\hat{u}_{i+1}$) are recovered by a change of basis: in finite precision, we have

$$\hat{v}_{i+1} = fl(\hat{\mathcal{Y}}_k \hat{v}_{k,j+1}') = \left(\hat{\mathcal{Y}}_k + \delta\hat{\mathcal{Y}}_{k,v_{j+1}}\right)\hat{v}_{k,j+1}', \quad |\delta\hat{\mathcal{Y}}_{k,v_{j+1}}| \leq \epsilon(2s+2)|\hat{\mathcal{Y}}_k|, \tag{4.21}$$

and

$$\hat{u}_{i+1} = fl(\hat{\mathcal{Y}}_k \hat{u}_{k,j+1}') = \left(\hat{\mathcal{Y}}_k + \delta\hat{\mathcal{Y}}_{k,u_{j+1}}\right)\hat{u}_{k,j+1}', \quad |\delta\hat{\mathcal{Y}}_{k,u_{j+1}}| \leq \epsilon(2s+2)|\hat{\mathcal{Y}}_k|. \tag{4.22}$$

We can now prove (4.8) in Theorem 4.2. Using (4.17)–(4.19) and (4.21), as well as the bounds in (4.14)–(4.16), (4.21), (4.22), and Lemma 4.1, we obtain

$$\left|\hat{v}_{i+1}^T \hat{v}_{i+1} - 1\right| \leq \epsilon(n+8s+12)\Gamma_k^2 \leq \epsilon_0/2, \tag{4.23}$$

where ϵ_0 is defined in (4.10). This thus satisfies the bound (4.8).

We now proceed toward proving (4.7). Line 12 in Algorithm 2 is computed as

$$\hat{\alpha}_i = fl(\hat{v}_{k,j}'^T \hat{G}_k \hat{u}'_{k,j}) = (\hat{v}_{k,j}'^T + \delta\hat{v}_{k,j}'^T)(\hat{G}_k + \delta\hat{G}_{k,u_j})\hat{u}'_{k,j},$$

where $|\delta\hat{v}'_{k,j}| \leq \epsilon(2s+2)|\hat{v}'_{k,j}|$ and $|\delta\hat{G}_{k,u_j}| \leq \epsilon(2s+2)|\hat{G}_k|$. Expanding the above equation using (4.14), we obtain

$$\hat{\alpha}_i = \hat{v}_{k,j}'^T \hat{\mathcal{Y}}_k^T \hat{\mathcal{Y}}_k \hat{u}'_{k,j} + \hat{v}_{k,j}'^T \delta G_k \hat{u}'_{k,j} + \hat{v}_{k,j}'^T \delta\hat{G}_{k,u_j} \hat{u}'_{k,j} + \delta\hat{v}_{k,j}'^T \hat{G}_k \hat{u}'_{k,j},$$

and since by (4.21) and (4.22), $\hat{v}_i^T \hat{u}_i = \hat{v}_{k,j}'^T \hat{\mathcal{Y}}_k^T \hat{\mathcal{Y}}_k \hat{u}'_{k,j} + \hat{\mathcal{Y}}_k^T \delta\hat{\mathcal{Y}}_{k,u_j} \hat{u}'_{k,j} + \delta\hat{\mathcal{Y}}_{k,v_j}^T \hat{\mathcal{Y}}_k \hat{u}'_{k,j}$, we can write

$$\hat{\alpha}_i = \hat{v}_i^T \hat{u}_i + \delta\hat{\alpha}_i, \tag{4.24}$$

where $\delta\hat{\alpha}_i = \delta\hat{v}_{k,j}'^T \hat{G}_k \hat{u}'_{k,j} + \hat{v}_{k,j}'^T (\delta G_k + \delta\hat{G}_{k,u_j} - \hat{\mathcal{Y}}_k^T \delta\hat{\mathcal{Y}}_{k,u_j} - \delta\hat{\mathcal{Y}}_{k,v_j}^T \hat{\mathcal{Y}}_k)\hat{u}'_{k,j}$. Using Lemma 4.1 along with the bounds in (4.14)–(4.15) and (4.21)–(4.23),

$$|\delta\hat{\alpha}_i| \leq \epsilon(n+8s+8)\Gamma_k^2 \big\|\hat{u}_i\big\|_2, \tag{4.25}$$

and then using (4.24) and the bounds in (4.23) and (4.25), we obtain

$$|\hat{\alpha}_i| \leq \Big(1 + \epsilon\big((3/2)n+12s+14\big)\Gamma_k^2\Big)\|\hat{u}_i\|_2. \tag{4.26}$$

In finite precision, line 13 of Algorithm 2 is computed as

$$\hat{w}'_{k,j} = \hat{u}'_{k,j} - \hat{\alpha}_i \hat{v}'_{k,j} - \delta w'_{k,j}, \quad \text{where} \quad |\delta w'_{k,j}| \leq \epsilon(|\hat{u}'_{k,j}| + 2|\hat{\alpha}_i \hat{v}'_{k,j}|). \tag{4.27}$$

Multiplying both sides of (4.27) by $\hat{\mathcal{Y}}_k$ gives $\hat{\mathcal{Y}}_k \hat{w}'_{k,j} = \hat{\mathcal{Y}}_k \hat{u}'_{k,j} - \hat{\alpha}_i \hat{\mathcal{Y}}_k \hat{v}'_{k,j} - \hat{\mathcal{Y}}_k \delta w'_{k,j}$, and multiplying each side by its own transpose and using (4.21) and (4.22),

$$\begin{aligned}\|\hat{\mathcal{Y}}_k \hat{w}'_{k,j}\|_2^2 = \|\hat{u}_i\|_2^2 - 2\hat{\alpha}_i \hat{u}_i^T \hat{v}_i + \hat{\alpha}_i^2 \|\hat{v}_i\|_2^2 \\ - 2(\delta\hat{\mathcal{Y}}_{k,u_j} \hat{u}'_{k,j} - \hat{\alpha}_i \delta\hat{\mathcal{Y}}_{k,v_j} \hat{v}'_{k,j} + \hat{\mathcal{Y}}_k \delta w'_{k,j})^T (\hat{u}_i - \hat{\alpha}_i \hat{v}_i),\end{aligned}$$

where we have used $\hat{\mathcal{Y}}_k \hat{u}'_{k,j} - \hat{\alpha}_i \hat{\mathcal{Y}}_k \hat{v}'_{k,j} = \hat{u}_i - \hat{\alpha}_i \hat{v}_i + O(\epsilon)$. Now, using (4.24) and rearranging the above, we obtain

$$\begin{aligned}\|\hat{\mathcal{Y}}_k \hat{w}'_{k,j}\|_2^2 + \hat{\alpha}_i^2 - \|\hat{u}_i\|_2^2 = \hat{\alpha}_i^2(\|\hat{v}_i\|_2^2 - 1) + 2\hat{\alpha}_i \delta\hat{\alpha}_i \\ - 2(\delta\hat{\mathcal{Y}}_{k,u_j} \hat{u}'_{k,j} - \hat{\alpha}_i \delta\hat{\mathcal{Y}}_{k,v_j} \hat{v}'_{k,j} + \hat{\mathcal{Y}}_k \delta w'_{k,j})^T (\hat{u}_i - \hat{\alpha}_i \hat{v}_i).\end{aligned}$$

Using Lemma 4.1 and the bounds in (4.21), (4.22), (4.23), (4.25), (4.26), and (4.27), we can then write the bound

$$\|\hat{\mathcal{Y}}_k \hat{w}'_{k,j}\|_2^2 + \hat{\alpha}_i^2 - \|\hat{u}_i\|_2^2 \leq \epsilon(3n + 40s + 56)\Gamma_k^2 \|\hat{u}_i\|_2^2. \tag{4.28}$$

Given the above, we can also write

$$\|\hat{\mathcal{Y}}_k \hat{w}'_{k,j}\|_2^2 \leq \|\hat{\mathcal{Y}}_k \hat{w}'_{k,j}\|_2^2 + \hat{\alpha}_i^2 \leq \big(1 + \epsilon(3n + 40s + 56)\Gamma_k^2\big)\|\hat{u}_i\|_2^2, \tag{4.29}$$

and using this with (4.16), (4.17), and (4.18),

$$|\hat{\beta}_{i+1}| \leq \big(1 + \epsilon(2n+22s+31)\Gamma_k^2\big)\big\|\hat{u}_i\big\|_2. \tag{4.30}$$

Now, rearranging (4.19) and using (4.21), we can write

$$\hat{\beta}_{i+1}\hat{v}_{i+1} \equiv \hat{\mathcal{Y}}_k \hat{w}'_{k,j} + \delta w_i, \tag{4.31}$$

where $\delta w_i = \hat{\mathcal{Y}}_k \delta\tilde{w}'_{k,j} + \delta\hat{\mathcal{Y}}_{k,v_{j+1}} \hat{w}'_{k,j}$, and using Lemma 4.1 and the bounds in (4.20), (4.21), and (4.29),

$$\|\delta w_i\|_2 \leq \epsilon(2s+3)\Gamma_k \|\hat{u}_i\|_2. \tag{4.32}$$

We premultiply (4.31) by $\hat{v}_i^T$ and use (4.21), (4.22), (4.24), and (4.27) to obtain

$$\hat{\beta}_{i+1}\hat{v}_i^T\hat{v}_{i+1} = -\delta\hat{\alpha}_i - \hat{\alpha}_i(\|\hat{v}_i\|_2^2 - 1) - \hat{v}_i^T(\delta\hat{\mathcal{Y}}_{k,u_j}\hat{u}'_{k,j} - \hat{\alpha}_i\delta\hat{\mathcal{Y}}_{k,v_j}\hat{v}'_{k,j} + \hat{\mathcal{Y}}_k\delta w'_{k,j} - \delta w_i),$$

and using Lemma 4.1 together with the bounds in (4.23), (4.19), (4.21), (4.22), (4.25), (4.26), (4.27), and (4.32), we can write the bound

$$\left|\hat{\beta}_{i+1}\hat{v}_i^T\hat{v}_{i+1}\right| \leq 2\epsilon(n+11s+15)\Gamma_k^2\|\hat{u}_i\|_2. \tag{4.33}$$

This is a start toward proving (4.7). We will return to the above bound once we later prove a bound on $\|\hat{u}_i\|_2$. Our next step is to analyze the error in each column of the finite precision s-step Lanczos recurrence. First, we note that we can write the finite precision recurrence for computing the s-step bases (line 3 in Algorithm 2) as

$$A\underline{\hat{\mathcal{Y}}}_k = \hat{\mathcal{Y}}_k\mathcal{B}_k + \delta E_k. \tag{4.34}$$

If the basis is computed by repeated matrix-vector products,

$$|\delta E_k| \leq \epsilon\big((3+N)|A||\underline{\hat{\mathcal{Y}}}_k| + 4|\hat{\mathcal{Y}}_k||\mathcal{B}_k|\big), \tag{4.35}$$

where N is the maximum number of nonzeros per row over all rows of A (see, e.g., [6]).

In finite precision, line 17 in Algorithm 2 is computed as

$$\hat{u}'_{k,j} = \mathcal{B}_k\hat{v}'_{k,j} - \hat{\beta}_i\hat{v}'_{k,j-1} + \delta u'_{k,j},\ |\delta u'_{k,j}| \leq \epsilon\big((2s+3)|\mathcal{B}_k||\hat{v}'_{k,j}| + 2|\hat{\beta}_i\hat{v}'_{k,j-1}|\big). \tag{4.36}$$

Then with Lemma 4.1, (4.21), (4.22), (4.34), and (4.36), we can write

$$\hat{u}_i = A\hat{v}_i - \hat{\beta}_i\hat{v}_{i-1} + \delta u_i, \tag{4.37}$$

where

$$\delta u_i = \hat{\mathcal{Y}}_k\delta u'_{k,j} - (A\delta\hat{\mathcal{Y}}_{k,v_j} - \delta\hat{\mathcal{Y}}_{k,u_j}\mathcal{B}_k + \delta E_k)\hat{v}'_{k,j} + \hat{\beta}_i(\delta\hat{\mathcal{Y}}_{k,v_{j-1}} - \delta\hat{\mathcal{Y}}_{k,u_j})\hat{v}'_{k,j-1},$$

and with the bounds in (4.21), (4.22), (4.30), (4.35), and (4.36),

$$\|\delta u_i\|_2 \leq \epsilon(N+2s+5)\,\||A|\|_2\Gamma_k + \epsilon(4s+9)\,\||\mathcal{B}_k|\|_2\Gamma_k + \epsilon(4s+6)\,\|\hat{u}_{i-1}\|_2\Gamma_k.$$

We will now introduce and make use of the quantities $\sigma \equiv \|A\|_2$, $\theta \equiv \||A|\|_2/\sigma$, and $\tau_k \equiv \||\mathcal{B}_k|\|_2/\sigma$. Note that the quantity $\||\mathcal{B}_k|\|_2$ depends on the choice of polynomials used in constructing the Krylov bases. For the monomial basis, $\||\mathcal{B}_k|\|_2 = 1$. For bases based on the spectrum of A, including Newton and Chebyshev bases, we expect that $\||\mathcal{B}_k|\|_2 \lesssim \||A|\|_2$ as long as the bases are constructed using sufficiently accurate spectral estimates. Using this notation, the bound above can be written

$$\big\|\delta u_i\big\|_2 \leq \epsilon\Big(\big((N+2s+5)\theta + (4s+9)\tau_k\big)\sigma + (4s+6)\|\hat{u}_{i-1}\|_2\Big)\Gamma_k. \tag{4.38}$$

Now, manipulating (4.31) with (4.21), (4.22), and (4.27), we have

$$\hat{\beta}_{i+1}\hat{v}_{i+1} = \hat{u}_i - \hat{\alpha}_i\hat{v}_i - \delta\hat{\mathcal{Y}}_{k,u_j}\hat{u}'_{k,j} + \hat{\alpha}_i\delta\hat{\mathcal{Y}}_{k,v_j}\hat{v}'_{k,j} - \hat{\mathcal{Y}}_k\delta w'_{k,j} + \delta w_i,$$

and substituting in the expression for $\hat{u}_i$ in (4.37),

$$\hat{\beta}_{i+1}\hat{v}_{i+1} = A\hat{v}_i - \hat{\alpha}_i\hat{v}_i - \hat{\beta}_i\hat{v}_{i-1} + \delta\hat{v}_i, \tag{4.39}$$

where $\delta\hat{v}_i = \delta u_i - \delta\hat{\mathcal{Y}}_{k,u_j}\hat{u}'_{k,j} + \hat{\alpha}_i\delta\hat{\mathcal{Y}}_{k,v_j}\hat{v}'_{k,j} - \hat{\mathcal{Y}}_k\delta w'_{k,j} + \delta w_i$. Using Lemma 4.1, along with (4.20), (4.21), (4.22), (4.26), (4.27), and (4.32),

$$\|\delta\hat{v}_i\|_2 \le \|\delta u_i\|_2 + \epsilon(6s+10)\Gamma_k\|\hat{u}_i\|_2,$$

and using (4.38), this bound becomes

$$\|\delta\hat{v}_i\|_2 \le \epsilon\Big(\big((N+2s+5)\theta + (4s+9)\tau_k\big)\sigma + (6s+10)\|\hat{u}_i\|_2 + (4s+6)\|\hat{u}_{i-1}\|_2\Big)\Gamma_k. \tag{4.40}$$

We now have everything we need to write the finite-precision s-step Lanczos recurrence in its familiar matrix form. Let $\hat{V}_i$, $\delta\hat{V}_i$, and $\hat{T}_i$ be defined as in (4.3), (4.4), and (4.5), respectively. Then (4.39) in matrix form gives

$$A\hat{V}_i = \hat{V}_i\hat{T}_i + \hat{\beta}_{i+1}\hat{v}_{i+1}e_i^T - \delta\hat{V}_i. \tag{4.41}$$

Thus, (4.40) gives a bound on the error in the columns of the finite precision s-step Lanczos recurrence. We will return to (4.40) to prove (4.6) once we bound $\|\hat{u}_i\|_2$.

Now, we examine the possible loss of orthogonality in the vectors $\hat{v}_1, \ldots, \hat{v}_{i+1}$. We define the strictly upper triangular matrix R_i of dimension i-by-i ($(sk+j)$-by-$(sk+j)$) with elements $\rho_{\ell,t}$ for $\ell, t \in \{1, \ldots, i\}$, such that $\hat{V}_i^T\hat{V}_i = R_i^T + \mathrm{diag}(\hat{V}_i^T\hat{V}_i) + R_i$. For notational purposes, we also define $\rho_{i,i+1} \equiv \hat{v}_i^T\hat{v}_{i+1}$. Multiplying (4.41) on the left by $\hat{V}_i^T$ and equating the right-hand side by its own transpose, we obtain

$$\begin{aligned}\hat{T}_i(R_i^T + R_i) - (R_i^T + R_i)\hat{T}_i = \hat{\beta}_{i+1}(\hat{V}_i^T\hat{v}_{i+1}e_i^T - e_i\hat{v}_{i+1}^T\hat{V}_i) + \hat{V}_i^T\delta\hat{V}_i - \delta\hat{V}_i^T\hat{V}_i \\ + \mathrm{diag}(\hat{V}_i^T\hat{V}_i)\cdot\hat{T}_i - \hat{T}_i\cdot\mathrm{diag}(\hat{V}_i^T\hat{V}_i).\end{aligned}$$

Now, let $M_i \equiv \hat{T}_iR_i - R_i\hat{T}_i$, which is upper triangular and has dimension i-by-i. By definition,

$$\begin{aligned} m_{1,1} &= -\hat{\beta}_2\rho_{1,2}, \qquad m_{i,i} = \hat{\beta}_i\rho_{i-1,i}, \quad \text{and} \\ m_{\ell,\ell} &= \hat{\beta}_\ell\rho_{\ell-1,\ell} - \hat{\beta}_{\ell+1}\rho_{\ell,\ell+1} \quad \text{for} \quad \ell \in \{2, \ldots, i-1\}.\end{aligned}$$

Therefore, $M_i = \hat{\beta}_{i+1}\hat{V}_i^T\hat{v}_{i+1}e_i^T + \delta R_i$, where δR_i has elements satisfying

$$\begin{aligned} \eta_{1,1} &= -\hat{\beta}_2\rho_{1,2}, \quad \text{and for } \ell \in \{2, \ldots, i\}, \\ \eta_{\ell,\ell} &= \hat{\beta}_\ell\rho_{\ell-1,\ell} - \hat{\beta}_{\ell+1}\rho_{\ell,\ell+1}, \\ \eta_{\ell-1,\ell} &= \hat{v}_{\ell-1}^T\delta\hat{v}_\ell - \delta\hat{v}_{\ell-1}^T\hat{v}_\ell + \hat{\beta}_\ell(\hat{v}_{\ell-1}^T\hat{v}_{\ell-1} - \hat{v}_\ell^T\hat{v}_\ell), \ \text{and} \\ \eta_{t,\ell} &= \hat{v}_t^T\delta\hat{v}_\ell - \delta\hat{v}_t^T\hat{v}_\ell, \quad \text{where } t \in \{1, \ldots, \ell-2\}.\end{aligned} \tag{4.42}$$

To simplify notation, we introduce the quantities

$$\bar{u}_i = \max_{\ell\in\{1,\ldots,i\}}\|\hat{u}_\ell\|_2, \quad \bar{\Gamma}_k = \max_{\ell\in\{0,\ldots,k\}}\Gamma_\ell, \quad \text{and} \quad \bar{\tau}_k = \max_{\ell\in\{0,\ldots,k\}}\tau_\ell.$$

Using this notation and (4.23), (4.30), (4.33), and (4.40), the quantities in (4.42) can be bounded by

(4.43)
$$\begin{aligned}
|\eta_{1,1}| &\le 2\epsilon(n+11s+15)\bar{\Gamma}_k^2\bar{u}_i, \text{ and for } \ell \in \{2,\dots,i\},\\
|\eta_{\ell,\ell}| &\le 4\epsilon(n+11s+15)\bar{\Gamma}_k^2\bar{u}_i,\\
|\eta_{\ell-1,\ell}| &\le 2\epsilon\Big(\big((N+2s+5)\theta+(4s+9)\bar{\tau}_k\big)\sigma+(10s+16)\bar{u}_i\Big)\bar{\Gamma}_k+\Big(n+8s+12\Big)\bar{\Gamma}_k^2\bar{u}_i\Big),\\
|\eta_{t,\ell}| &\le 2\epsilon\Big(\big((N+2s+5)\theta+(4s+9)\bar{\tau}_k\big)\sigma+(10s+16)\bar{u}_i\Big)\bar{\Gamma}_k, \text{ where } t \in \{1,\dots,\ell-2\}.
\end{aligned}$$

The above is a start toward proving (4.13). We return to this bound later and now shift our focus toward proving a bound on $\|\hat{u}_i\|_2$. To proceed, we must first find a bound for $|\rho_{i-2,i}|$. We know that the $(1,2)$ element of M_i is

$$\eta_{1,2} = \hat{\alpha}_1\rho_{1,2} - \hat{\alpha}_2\rho_{1,2} - \hat{\beta}_3\rho_{1,3},$$

and for $\ell > 2$, the $(\ell-1,\ell)$ element is

$$\eta_{\ell-1,\ell} = \hat{\beta}_{\ell-1}\rho_{\ell-2,\ell} + (\hat{\alpha}_{\ell-1} - \hat{\alpha}_\ell)\rho_{\ell-1,\ell} - \hat{\beta}_{\ell+1}\rho_{\ell-1,\ell+1}.$$

Then, defining $\xi_\ell \equiv (\hat{\alpha}_{\ell-1} - \hat{\alpha}_\ell)\hat{\beta}_\ell\rho_{\ell-1,\ell} - \hat{\beta}_\ell\eta_{\ell-1,\ell}$ for $\ell \in \{2,\dots,i\}$, we have

$$\hat{\beta}_\ell\hat{\beta}_{\ell+1}\rho_{\ell-1,\ell+1} = \hat{\beta}_{\ell-1}\hat{\beta}_\ell\rho_{\ell-2,\ell} + \xi_\ell = \xi_\ell + \xi_{\ell-1} + \ldots + \xi_2.$$

This, along with (4.26), (4.30), (4.33), and (4.43), gives

$$\begin{aligned}
\hat{\beta}_i\hat{\beta}_{i+1}|\rho_{i-1,i+1}| &\le \sum_{\ell=2}^{i}|\xi_\ell| \le \sum_{\ell=2}^{i}(|\hat{\alpha}_{\ell-1}| + |\hat{\alpha}_\ell|)|\hat{\beta}_\ell\rho_{\ell-1,\ell}| + |\hat{\beta}_\ell||\eta_{\ell-1,\ell}|\\
&\le 2\epsilon(i-1)\Big(\big((N+2s+5)\theta + (4s+9)\bar{\tau}_k\big)\sigma + (10s+16)\bar{u}_i\Big)\bar{\Gamma}_k\bar{u}_i\\
&\quad + 2\epsilon(i-1)\cdot 3\big(n+10s+14\big)\bar{\Gamma}_k^2\bar{u}_i^2. \qquad (4.44)
\end{aligned}$$

Rearranging (4.37) gives $\hat{u}_i - \delta u_i = A\hat{v}_i - \hat{\beta}_i\hat{v}_{i-1}$, and multiplying each side by its own transpose (and ignoring all terms of second order in ϵ), we obtain

$$\hat{u}_i^T\hat{u}_i - 2\hat{u}_i^T\delta u_i = \|A\hat{v}_i\|_2^2 + \hat{\beta}_i^2\|\hat{v}_{i-1}\|_2^2 - 2\hat{\beta}_i\hat{v}_i^T A\hat{v}_{i-1}. \qquad (4.45)$$

Rearranging (4.39) gives $A\hat{v}_{i-1} = \hat{\beta}_i\hat{v}_i + \hat{\alpha}_{i-1}\hat{v}_{i-1} + \hat{\beta}_{i-1}\hat{v}_{i-2} - \delta\hat{v}_{i-1}$, and premultiplying this expression by $\hat{\beta}_i\hat{v}_i^T$, we get

$$\hat{\beta}_i\hat{v}_i^T A\hat{v}_{i-1} = \hat{\beta}_i^2 + \delta\hat{\beta}_i, \qquad (4.46)$$

where, using the bounds in (4.23), (4.26), (4.30), (4.33), (4.40), and (4.44),

$$\begin{aligned}
|\delta\hat{\beta}_i| &\le \epsilon\big(2i-1\big)\Big(\big((N+2s+5)\theta + (4s+9)\bar{\tau}_k\big)\sigma + (10s+16)\bar{u}_i\Big)\bar{\Gamma}_k\bar{u}_i\\
&\quad + \epsilon\big(2i-1\big)\cdot 3\big(n+10s+14\big)\bar{\Gamma}_k^2\bar{u}_i^2. \qquad (4.47)
\end{aligned}$$

Adding $2\hat{u}_i^T\delta u_i$ to both sides of (4.45) and using (4.46), we obtain

$$\|\hat{u}_i\|_2^2 = \|A\hat{v}_i\|_2^2 + \hat{\beta}_i^2\big(\|\hat{v}_{i-1}\|_2^2 - 2\big) + \delta\tilde{\beta}_i, \qquad (4.48)$$

where $\delta\tilde{\beta}_i = -2\delta\hat{\beta}_i + 2\hat{u}_i^T\delta u_i$, and, using the bounds in (4.38) and (4.47),

$$|\delta\tilde{\beta}_i| \leq 4i\epsilon\big((N+2s+5)\theta + (4s+9)\bar{\tau}_k\big)\sigma\bar{\Gamma}_k\bar{u}_i + 2\epsilon\Big(\big(2i-1\big)\big(3(n+10s+14)\bar{\Gamma}_k^2 + (10s+16)\bar{\Gamma}_k\big) + (4s+6)\bar{\Gamma}_k\Big)\bar{u}_i^2. \tag{4.49}$$

Now, using (4.48), and since $\hat{\beta}_i^2 \geq 0$, we can write

$$\|\hat{u}_i\|_2^2 \leq \|\hat{u}_i\|_2^2 + \hat{\beta}_i^2 \leq \sigma^2\|\hat{v}_i\|_2^2 + \hat{\beta}_i^2\big(\|\hat{v}_{i-1}\|_2^2 - 1\big) + |\delta\tilde{\beta}_i|. \tag{4.50}$$

Let $\mu \equiv \max\big\{\bar{u}_i, \sigma\big\}$. We can now put the bounds in terms of ϵ_0 and ϵ_1, which are defined in (4.10). Then, using (4.50), along with the bounds in (4.23), (4.30), and (4.49),

$$\|\hat{u}_i\|_2^2 \leq \sigma^2 + 2i(3\epsilon_0 + 2\epsilon_1)\mu^2. \tag{4.51}$$

We consider the two possible cases for μ. First, if $\mu = \sigma$, then

$$\|\hat{u}_i\|_2^2 \leq \sigma^2\left(1 + 2i(3\epsilon_0 + 2\epsilon_1)\right).$$

Otherwise, we have the case $\mu = \bar{u}_i$. Since the bound in (4.51) holds for all $\|\hat{u}_i\|_2^2$, it also holds for $\bar{u}_i^2 = \mu^2$, and thus, ignoring terms of second order in ϵ,

$$\mu^2 \leq \sigma^2 + 2i(3\epsilon_0 + 2\epsilon_1)\mu^2 \leq \sigma^2 + 2i(3\epsilon_0 + 2\epsilon_1)\sigma^2 \leq \sigma^2\left(1 + 2i(3\epsilon_0 + 2\epsilon_1)\right),$$

and, plugging this in to (4.51), we get

$$\|\hat{u}_i\|_2^2 \leq \sigma^2\left(1 + 2i(3\epsilon_0 + 2\epsilon_1)\right). \tag{4.52}$$

In either case, we obtain the same bound on $\|\hat{u}_i\|_2^2$, so (4.52) holds.

Taking the square root of (4.52), we have

$$\|\hat{u}_i\|_2 \leq \sigma\left(1 + i(3\epsilon_0 + 2\epsilon_1)\right), \tag{4.53}$$

and substituting (4.53) into (4.33), (4.40), and (4.43), we prove the bounds (4.7), (4.6), and (4.13) in Theorem 4.2, respectively, assuming that $i(3\epsilon_0 + 2\epsilon_1) \ll 1$.

The only remaining inequality to prove is (4.9). We first multiply both sides of (4.31) by their own transposes and then add $\hat{\alpha}_i^2 - \|\hat{u}_i\|_2^2$ to both sides to obtain

$$\hat{\beta}_{i+1}^2\|\hat{v}_{i+1}\|_2^2 + \hat{\alpha}_i^2 - \|\hat{u}_i\|_2^2 = \|\hat{\mathcal{Y}}_k\hat{w}'_{k,j}\|_2^2 + \hat{\alpha}_i^2 - \|\hat{u}_i\|_2^2 + 2\delta w_i^T\hat{\mathcal{Y}}_k\hat{w}'_{k,j}.$$

We then substitute in (4.48) on the left-hand side, subtract $\hat{\beta}_{i+1}^2$ from both sides, and rearrange to obtain

$$\hat{\beta}_{i+1}^2 + \hat{\alpha}_i^2 + \hat{\beta}_i^2 - \|A\hat{v}_i\|_2^2 = \|\hat{\mathcal{Y}}_k\hat{w}'_{k,j}\|_2^2 + \hat{\alpha}_i^2 - \|\hat{u}_i\|_2^2 + 2\delta w_i^T\hat{\mathcal{Y}}_k\hat{w}'_{k,j} + \hat{\beta}_i^2(\|\hat{v}_{i-1}\|_2^2 - 1) - \hat{\beta}_{i+1}^2(\|\hat{v}_{i+1}\|_2^2 - 1) + \delta\tilde{\beta}_i.$$

Finally, using (4.23), (4.28), (4.30), (4.32), and (4.49), we arrive at the bound

$$\left|\hat{\beta}_{i+1}^2 + \hat{\alpha}_i^2 + \hat{\beta}_i^2 - \|A\hat{v}_i\|_2^2\right| \leq 2i(3\epsilon_0 + 2\epsilon_1)\sigma^2,$$

which proves (4.9) and thus completes the proof of Theorem 4.2.

5. Accuracy of eigenvalues. Theorem 4.2 is in the same form as Paige's equivalent theorem for classical Lanczos [36], except our definitions of ϵ_0 and ϵ_1 are about a factor $\bar{\Gamma}_k^2$ larger (assuming $s \ll n$). This amplification term, which can be bounded in terms of the maximum condition number of the computed s-step Krylov bases, has significant consequences for the algorithm, as we will see in the next two sections. The equivalent forms of our theorem and Paige's theorem allow us to immediately apply his results from [36] to the s-step case; the only thing that changes in the s-step case are the values of ϵ_0 and ϵ_1.

In this and the subsequent section, we reproduce the theorems of Paige and discuss their application to the s-step Lanczos method. *Note that the present authors claim no contribution to the analysis techniques used here.* In fact, much of the text in the following sections is taken verbatim from Paige [36], with only the notation changed to match the algorithms in section 3.

Our contribution is showing that the theorems of Paige also apply to the s-step Lanczos method under the assumption that (5.1) (and thus also (5.2)) holds. Also note that the text in the paragraphs labeled "Comments," which discusses the meaning of the results for the s-step case, is our own. Again, to simplify the exposition, we have omitted proofs of the theorems stated here which are due to Paige [36]. For the unfamiliar reader, a much longer manuscript on the s-step analysis including proofs and intermediate steps omitted by Paige can be found in the technical report [3].

We note that many of the bounds stated slightly differ from those given by Paige in [36]. The present authors suspect that the bounds in [36] were obtained using $\epsilon_0 < 1/100$ rather than the specified $\epsilon_0 < 1/12$, the former being the value used by Paige in his earlier work [33]. Such changes are indicated by footnotes and carried through the remainder of the analysis, resulting in different constants than those in [36]; the fundamental results and conclusions remain unchanged.

Assumptions. In order to make use of Paige's analysis [36], we must make the similar assumptions that

$$\hat{\beta}_{i+1} \neq 0 \ \text{ for } \ i \in \{1, \ldots, m\}, \quad m(3\epsilon_0 + 2\epsilon_1) \leq 1, \ \text{ and } \epsilon_0 < \frac{1}{12}. \tag{5.1}$$

These assumptions are used throughout the analysis. Note that (5.1) means that in order to guarantee the applicability of Paige's results for classical Lanczos to the s-step Lanczos case, we must have

$$\bar{\Gamma}_k^2 < \Big(24\epsilon(n+11s+15)\Big)^{-1} = O\big(1/(n\epsilon)\big). \tag{5.2}$$

Since the bounds that will be presented, as well as the bounds in Theorem 4.2, are not tight, this condition on $\bar{\Gamma}_k^2$ may be overly restrictive in practice. In paragraphs labeled "Comments," we comment on what happens to the bounds and analysis in the case that $\bar{\Gamma}_k^2$ exceeds this value, i.e., at least one computed s-step basis is ill-conditioned. As stated previously, we also assume that no underflow or overflow occurs and that all s-step Krylov bases are numerically full rank.

Using (4.13), it can be shown that

$$\|\delta R_m\|_F^2 \leq \sigma^2 \Big(\big(5m-4\big)\epsilon_0^2 + 4(m-1)\epsilon_0\epsilon_1 + 2m(m-1)\epsilon_1^2 \Big), \tag{5.3}$$

where subscript F denotes the Frobenius norm. If we define

$$\epsilon_2 \equiv \sqrt{2}\max\left(6\epsilon_0, \epsilon_1\right), \tag{5.4}$$

then (5.3) gives

$$\|\delta R_m\|_F \le m\sigma\epsilon_2. \tag{5.5}$$

Let the eigendecomposition of $\hat{T}_m$ be

$$\hat{T}_m Q^{(m)} = Q^{(m)} \operatorname{diag}\big(\mu_i^{(m)}\big) \tag{5.6}$$

for $i \in \{1, \ldots, m\}$, where the orthonormal matrix $Q^{(m)}$ has ith column $q_i^{(m)}$ and (ℓ, i) element $\eta_{\ell,i}^{(m)}$, and the eigenvalues are ordered

$$\mu_1^{(m)} > \mu_2^{(m)} > \cdots > \mu_m^{(m)}.$$

Note that it is assumed that the decomposition (5.6) is computed exactly. If $\mu_i^{(m)}$ is an approximation to an eigenvalue λ_i of A, then the corresponding approximate eigenvector is $z_i^{(m)}$, the ith column of

$$Z^{(m)} \equiv \hat{V}_m Q^{(m)}. \tag{5.7}$$

We now review some properties of $\hat{T}_m$. Let $\nu_i^{(m)}$, for $i \in \{1, \ldots, m-1\}$, be the eigenvalues of the matrix obtained by removing the $(t+1)$st row and column of $\hat{T}_m$, ordered so that

$$\mu_1^{(m)} \ge \nu_1^{(m)} \ge \mu_2^{(m)} \ge \cdots \ge \nu_{m-1}^{(m)} \ge \mu_m^{(m)}.$$

It was shown in [44] that

$$\left(\eta_{t+1,i}^{(m)}\right)^2 = \prod_{\ell=1,\ell\neq i}^{m} \delta_\ell(t+1, i, m), \tag{5.8}$$

$$\delta_\ell(t+1, i, m) \equiv \begin{cases} \frac{\mu_i^{(m)} - \nu_\ell^{(m)}}{\mu_i^{(m)} - \mu_\ell^{(m)}}, & \ell = 1, 2, \ldots, i-1, \\ \frac{\mu_i^{(m)} - \nu_{\ell-1}^{(m)}}{\mu_i^{(m)} - \mu_\ell^{(m)}}, & \ell = i+1, \ldots, m, \end{cases}$$

$$0 \le \delta_\ell(t+1, i, m) \le 1, \ell = 1, \ldots, i-1, i+1, \ldots, m. \tag{5.9}$$

If we apply $\hat{T}_m$ to the rth eigenvector of $\hat{T}_t$, where $1 \le r \le t < m$,

$$\hat{T}_m \begin{bmatrix} q_r^{(t)} \\ 0_{m-t,1} \end{bmatrix} = \begin{bmatrix} \mu_r^{(t)} q_r^{(t)} \\ \hat{\beta}_{t+1} \eta_{t,r}^{(t)} e_1 \end{bmatrix}, \tag{5.10}$$

and from [49],

$$\delta_{t,r} \equiv \hat{\beta}_{t+1} |\eta_{t,r}^{(t)}| \ge \min_i |\mu_i^{(m)} - \mu_r^{(t)}|. \tag{5.11}$$

DEFINITION 5.1 (see [36, Definition 1]). *We say that an eigenvalue $\mu_r^{(t)}$ of $\hat{T}_t$* has stabilized to within *$\delta_{t,r}$ if, for every $m > t$, we know there is an eigenvalue of $\hat{T}_m$ within $\delta_{t,r}$ of $\mu_r^{(t)}$. We will say $\mu_r^{(t)}$ has* stabilized *when we know it has stabilized to within $\gamma(m+1)^\omega \sigma\epsilon_2$, where γ and ω are small positive constants.*

From (5.11), we can see that after t steps, $\mu_r^{(t)}$ has necessarily stabilized to within $\delta_{t,r}$. Multiplying (5.10) by $q_i^{(m)T}$, $i \in \{1, \ldots, m\}$, gives

$$\big(\mu_i^{(m)} - \mu_r^{(t)}\big) q_i^{(m)T} \begin{bmatrix} q_r^{(t)} \\ 0_{m-t,1} \end{bmatrix} = \hat{\beta}_{t+1} \eta_{t+1,i}^{(m)} \eta_{t,r}^{(t)}. \tag{5.12}$$

Another result is obtained by applying eigenvectors of $\hat{T}_m$ to each side of (4.12). Multiplying (4.12) on the left by $q_\ell^{(m)T}$ and on the right by $q_i^{(m)}$ for some $i, \ell \in \{1, \ldots, m\}$, and using (5.6) and (5.7), we obtain

$$\big(\mu_\ell^{(m)} - \mu_i^{(m)}\big) q_\ell^{(m)T} R_m q_i^{(m)} = \hat{\beta}_{m+1} z_\ell^{(m)T} \hat{v}_{m+1} \eta_{m,i}^{(m)} + \epsilon_{\ell,i}^{(m)}, \tag{5.13}$$

where $\epsilon_{\ell,i}^{(m)} \equiv q_\ell^{(m)T} \delta R_m q_i^{(m)}$, and

$$|\epsilon_{\ell,i}^{(m)}| \leq m\sigma\epsilon_2, \tag{5.14}$$

which follows from (5.5). Taking $i = \ell$, the left-hand side of (5.13) is zero, and we get

$$z_i^{(m)T} \hat{v}_{m+1} = -\frac{\epsilon_{i,i}^{(m)}}{\hat{\beta}_{m+1} \eta_{m,i}^{(m)}}, \tag{5.15}$$

and thus by (5.11), $z_i^{(m)}$ is almost orthogonal to $\hat{v}_{m+1}$ if we have not yet obtained a small eigenvalue interval about $\mu_i^{(m)}$, the eigenvector approximation $z_i^{(m)}$ does not have a small norm, and $\bar{\Gamma}_k$, and thus ϵ_0 and ϵ_2, are small.

DEFINITION 5.2 (see [36, Definition 2]). *We will say that an eigenpair (μ, z) represents an eigenpair of A to within δ if we know that $\|Az - \mu z\|/\|z\| \leq \delta$.*

Thus, if (μ, z) represents an eigenpair of A to within δ, then (μ, z) is an exact eigenpair of A perturbed by a matrix whose 2-norm is no greater than δ, and if μ is the Rayleigh quotient of A with z, then the perturbation will be taken symmetric.

Multiplying (4.2) on the right by $q_i^{(m)}$, we get

$$A\hat{V}_m q_i^{(m)} = \hat{V}_m \hat{T}_m q_i^{(m)} + \hat{\beta}_{m+1} \hat{v}_{m+1} e_m^T q_i^{(m)} + \delta \hat{V}_m q_i^{(m)}.$$

Using (5.6) and (5.7), this can be written

$$Az_i^{(m)} - \mu_i^{(m)} z_i^{(m)} = \hat{\beta}_{m+1} \eta_{m,i}^{(m)} \hat{v}_{m+1} + \delta \hat{V}_m q_i^{(m)}. \tag{5.16}$$

Now, using the above and (4.6), (4.8), and (5.11), if λ_ℓ are the eigenvalues of A, then

$$\min_\ell |\lambda_\ell - \mu_i^{(m)}| \leq \frac{\|Az_i^{(m)} - \mu_i^{(m)} z_i^{(m)}\|}{\|z_i^{(m)}\|} \leq \frac{\delta_{m,i}(1 + \epsilon_0) + m^{1/2}\sigma\epsilon_1}{\|z_i^{(m)}\|}, \tag{5.17}$$

and if

$$\|z_i^{(m)}\| \approx 1, \tag{5.18}$$

then $(\mu_i^{(m)}, z_i^{(m)})$ represents an eigenpair of A to within about $\delta_{m,i}$. Unfortunately, one can not expect (5.18) to hold in finite precision.

From (5.7) and (4.11), we see that

$$\|z_i^{(m)}\|^2 - 1 = 2q_i^{(m)T} R_m q_i^{(m)} + q_i^{(m)T} \mathrm{diag}(\hat{V}_m^T \hat{V}_m - I_m) q_i^{(m)}, \tag{5.19}$$

where by (4.8), the last term on the right has magnitude bounded by $\epsilon_0/2$.

Using (5.7), we can write $\hat{V}_t^T = Q^{(t)} Z^{(t)T}$, and multiplying on the right by $\hat{v}_{t+1}$,

$$\hat{V}_t^T \hat{v}_{t+1} = Q^{(t)} b_t, \quad \text{where } b_t = Z^{(t)T} \hat{v}_{t+1}. \tag{5.20}$$

Using (5.15), we have $e_r^T b_t = -\epsilon_{r,r}^{(t)}/(\hat{\beta}_{t+1}\eta_{t,r}^{(t)})$, and by (5.8) and (5.12), this gives

$$q_i^{(m)T} R_m q_i^{(m)} = -\sum_{t=1}^{m-1} \eta_{t+1,i}^{(m)} \sum_{r=1}^{t} \frac{\epsilon_{r,r}^{(t)}}{\hat{\beta}_{t+1}\eta_{t,r}^{(t)}} q_i^{(m)T} \begin{bmatrix} q_r^{(t)} \\ 0_{m-t,1} \end{bmatrix} \tag{5.21}$$

$$= -\sum_{t=1}^{m-1} \left(\eta_{t+1,i}^{(m)}\right)^2 \sum_{r=1}^{t} \frac{\epsilon_{r,r}^{(t)}}{\mu_i^{(m)} - \mu_r^{(t)}} \tag{5.22}$$

$$= -\sum_{t=1}^{m-1} \sum_{r=1}^{t} \left(\frac{\epsilon_{r,r}^{(t)}}{\mu_i^{(m)} - \mu_{c(r)}^{(m)}} \cdot \prod_{\substack{\ell=1 \\ \ell \neq i \\ \ell \neq c(r)}}^{m} \delta_\ell(t+1, i, m) \right). \tag{5.23}$$

From (5.19), under the assumptions in (5.1), $\|z_i^{(m)}\|$ will be significantly different from unity only if the right-hand sides of these last three numbered equations are large. In this case, (5.21) shows there must be a small $\delta_{t,r} = \hat{\beta}_{t+1}|\eta_{t,r}^{(t)}|$, and some $\mu_r^{(t)}$ has therefore stabilized. Equation (5.22) shows that some $\mu_r^{(t)}$ must be close to $\mu_i^{(m)}$, and combining this with (5.21), we will show that at least one such $\mu_r^{(t)}$ has stabilized. Finally, from (5.23), we see that there is at least one $\mu_{c(r)}^{(m)}$ close to $\mu_i^{(m)}$, so that $\mu_i^{(m)}$ cannot be a well-separated eigenvalue of $\hat{T}_m$. Conversely, if $\mu_i^{(m)}$ is a well-separated eigenvalue of $\hat{T}_m$, then (5.18) holds, and if $\mu_i^{(m)}$ has stabilized, then it and $z_i^{(m)}$ are a satisfactory approximation to an eigenpair of A.

Note that if the assumptions in (5.1) do not hold, $\|z_i^{(m)}\|$ can be significantly differ from unity if $|q_i^{(m)T} R_m q_i^{(m)}|$ is large or if $\epsilon_0/2$ is large (e.g., due to a large $\bar{\Gamma}_k^2$; see (4.10)). If $\|z_i^{(m)}\|$ is significantly different from unity and $\epsilon_0/2$ is large, we cannot necessarily draw meaningful conclusions about the eigenvalues of $\hat{T}_m$ via (5.21), (5.22), and (5.23) based on the size of $\|z_i^{(m)}\|$.

We will now quantify these claims. We first seek to obtain an upper bound on $|q_i^{(m)T} R_m q_i^{(m)}|$. From (5.5) and (5.14),

$$\sum_{r=1}^{t} \left(\epsilon_{r,r}^{(t)}\right)^2 \leq \sum_{r=1}^{t} \sum_{c=1}^{t} \left(\epsilon_{r,c}^{(t)}\right)^2 = \|\delta R_t\|_F^2 \leq t^2 \sigma^2 \epsilon_2^2,$$

and using the Cauchy–Schwarz inequality,

$$\left(\sum_{r=1}^{t} |\epsilon_{r,r}^{(t)}| \right)^2 \leq \sum_{r=1}^{t} \left(\epsilon_{r,r}^{(t)}\right)^2 \sum_{r=1}^{t} 1 \leq t^3 \sigma^2 \epsilon_2^2. \tag{5.24}$$

Similarly, using (5.23) and the bound in (5.9),

$$|q_i^{(m)T} R_m q_i^{(m)}| \leq \frac{m^{5/2} \sigma \epsilon_2}{(5/2) \min_{\ell \neq i} |\mu_i^{(m)} - \mu_\ell^{(m)}|}. \tag{5.25}$$

This bound is weak, but it shows that if

$$\min_{\ell \neq i} |\mu_i^{(m)} - \mu_\ell^{(m)}| \geq m^{5/2}\sigma\epsilon_2, \tag{5.26}$$

then from (5.25), $|q_i^{(m)T} R_m q_i^{(m)}| \leq 2/5$, and substituting this into (5.19),

$$\left| \|z_i^{(m)}\|^2 - 1 \right| \leq 2\left|q_i^{(m)T} R_m q_i^{(m)}\right| + \frac{\epsilon_0}{2} \leq \frac{4}{5} + \frac{\epsilon_0}{2}. \tag{5.27}$$

Thus, with the condition that $\epsilon_0 = 2\epsilon(n+11s+15)\bar{\Gamma}_k^2 < 1/12$ (see (5.1)), we can then guarantee that

$$0.39 < \|z_i^{(m)}\| < 1.4, \tag{5.28}$$ [1]

which has implications for (5.17).

Comments. Note that we could slightly loosen the bound (5.1) on ϵ_0 and still carry through much of the preceding analysis, although in (5.27) we have assumed that $\epsilon_0/2 < 1/5$. If we instead have $\epsilon_0/2 \geq 1/5$, we get the trivial bound $0 \leq \|z_i^{(m)}\|^2$. This bound is not useful because in the worst case, $z_i^{(m)}$ is the 0-vector, which indicates either breakdown of the method or rank-deficiency of some $\hat{\mathcal{Y}}_k$.

Note that from (5.7) and (4.8),

$$\left| \sum_{i=1}^{m} \|z_i^{(m)}\|_2^2 - m \right| \leq \frac{m\epsilon_0}{2}.$$

It was also proved in [33] that if $\mu_i^{(m)}, \ldots, \mu_{i+c}^{(m)}$ are $c+1$ eigenvalues of $\hat{T}_m$ which are close to each other but separate from the rest, then

$$\sum_{\ell=i}^{i+c} \|z_\ell^{(m)}\|^2 \approx c+1. \tag{5.29}$$

This means that it is possible to have several close eigenvalues of $\hat{T}_m$ corresponding to one simple eigenvalue of A. In this case, the columns of $Z_c \equiv [z_i^{(m)}, \ldots, z_{i+c}^{(m)}]$ will all correspond to one eigenvector z of A having $z^T z = 1$. We now state another result.

LEMMA 5.1. *Let $\hat{T}_m$ and $\hat{V}_m$ be the result of m steps of the s-step Lanczos method with (4.10) and (5.4), and let R_m be the strictly upper triangular matrix defined in (4.11). Then for each eigenpair $(\mu_i^{(m)}, q_i^{(m)})$ of $\hat{T}_m$, there exists a pair of integers (r,t) with $0 \leq r \leq t < m$ such that*

$$\delta_{t,r} \equiv \hat{\beta}_{t+1}|\eta_{t,r}^{(t)}| \leq \psi_{i,m} \quad \textit{and} \quad |\mu_i^{(m)} - \mu_r^{(t)}| \leq \psi_{i,m},$$

where

$$\psi_{i,m} \equiv \frac{m^2\sigma\epsilon_2}{\left|\sqrt{3}\, q_i^{(m)T} R_m q_i^{(m)}\right|}.$$

Proof. See [36, Lemma 3.1]. ∎

[1] Note that these bounds differ from those given by Paige in [36], which are $0.42 < \|z_i^{(m)}\| < 1.4$; the present authors suspect that the bounds in [36] were obtained using $\epsilon_0 < 1/100$ rather than the specified $\epsilon_0 < 1/12$, the former being the value used by Paige in his earlier work [33].

Comments. For classical Lanczos, these bounds show that if $\|z_i^{(m)}\|_2$ is significantly different from unity, then for some $t < m$ there is an eigenvalue of $\hat{T}_t$ which has stabilized and is close to $\mu_i^{(m)}$ [36]. For the s-step Lanczos case, the same holds with the assumptions in (5.1). These assumptions are necessary because otherwise, for s-step Lanczos, $\|z_i^{(m)}\|$ can significantly differ from unity if $|q_i^{(m)T} R_m q_i^{(m)}|$ is large or if $\epsilon_0/2$ is large (due to a large $\bar{\Gamma}_k^2$, see (4.10)). If $\|z_i^{(m)}\|$ is much different from unity and $\epsilon_0/2$ is large, we cannot necessarily say that there is an eigenvalue of $\hat{T}_t$ which has stabilized to within a meaningful bound even if $|q_i^{(m)T} R_m q_i^{(m)}|$ is small.

THEOREM 5.2. *If, with the conditions of Lemma* 5.1, *an eigenvalue* $\mu_i^{(m)}$ *of* $\hat{T}_m$ *produced by s-step Lanczos is stabilized so that*

$$\delta_{m,i} \equiv \hat{\beta}_{m+1} |\eta_{m,i}^{(m)}| \leq \sqrt{3} m^2 \sigma \epsilon_2, \tag{5.30}$$

and $\epsilon_0 < 1/12$, *then for some eigenvalue* λ_c *of* A,

$$|\lambda_c - \mu_i^{(m)}| \leq (m+1)^3 \sigma \epsilon_2. \tag{5.31}$$

Proof. See [36, Theorem 3.1]. ∎

Comments. As in [36], the bound (5.31) is not tight and thus should in no way be considered an indication of the maximum attainable accuracy. In practice, we can still observe convergence of the eigenvalues of T_m to eigenvalues of A with larger $\bar{\Gamma}_k^2$ than allowed by $\epsilon_0 < 1/12$. The constraint $\epsilon_0 \leq 1/12$ comes from the proof of the theorem above, which requires that $3/8 - \epsilon_0/2 \geq 1/3$. The present authors believe that the restriction on the size of ϵ_0 could be loosened by a constant factor by changing the form of the right-hand side of [36, equation 3.37] such that meaningful bounds are still obtained. This remains for future work.

The following shows that if (5.30) holds, we have an eigenvalue with a superior error bound to (5.31) and we also have a good eigenvector approximation.

COROLLARY 5.3. *If* (5.30) *holds, then for the final* (r,t) *pair in Theorem* 5.2, $(\mu_r^{(t)}, \hat{V}_t q_r^{(t)})$ *is an exact eigenpair for a matrix within* $6t^2\sigma\epsilon_2$ *of* A.

Proof. See [36, Corollary 3.1] and [3, Corollary 5.3]. ∎

As in the classical Lanczos case, the above is also the result we obtain for an eigenvalue of $\hat{T}_m$ produced by s-step Lanczos that is stabilized and well-separated.

Paige showed that one can also consider the accuracy of the $\mu_i^{(m)}$ as Rayleigh quotients [36]. With no rounding errors, $\mu_i^{(m)}$ is the Rayleigh quotient of A with $z_i^{(m)}$, and this gives the best bound from (5.16) and (5.17) with $\epsilon = 0$, i.e., in exact arithmetic. Here, (5.15) and (5.16) can be combined to give

$$z_i^{(m)T} A z_i^{(m)} - \mu_i^{(m)} z_i^{(m)T} z_i^{(m)} = -\epsilon_{i,i}^{(m)} + z_i^{(m)T} \delta \hat{V}_m q_i^{(m)}, \tag{5.32}$$

so if $\|z_i^{(m)}\| \approx 1$, then $\mu_i^{(m)}$ is close to the Rayleigh quotient

$$\varrho_i^{(m)} = z_i^{(m)T} A z_i^{(m)} / z_i^{(m)T} z_i^{(m)}.$$

If (5.26) holds, then $\|z_i^{(m)}\| > 0.39$, and thus using (5.32) and the bounds in (4.6), (5.4), and (5.14),

$$|\varrho_i^{(m)} - \mu_i^{(m)}| \leq 9 m \sigma \epsilon_2.$$

If $\|z_i^{(m)}\|$ is small, then it is unlikely that $\mu_i^{(m)}$ will be very close to $\varrho_i^{(m)}$, since a small $z_i^{(m)}$ will probably be inaccurate due to rounding errors. Equation (5.29) suggests that at least one of a group of close eigenvalues will have corresponding $\|z_i^{(m)}\| \gtrsim 1$. In fact, using (5.22), (5.24), and an argument similar to that used in Theorem 5.2, it can be shown that every $\mu_i^{(m)}$ lies within $m^{5/2}\sigma\epsilon_2$ of a Rayleigh quotient of A, and so with (4.10) and (5.4), all the $\mu_i^{(m)}$ lie in the interval

$$\lambda_{\min} - m^{5/2}\sigma\epsilon_2 \leq \mu_i^{(m)} \leq \lambda_{\max} + m^{5/2}\sigma\epsilon_2.$$

This differs from the bound on the distance of $\mu_i^{(m)}$ from an eigenvalue of A in (5.31), which requires that $\mu_i^{(m)}$ has stabilized.

We emphasize that whatever the size of $\delta_{m,i}$, the eigenvalue $\mu_i^{(m)}$ of $\hat{T}_m$ with eigenvector $q_i^{(m)}$ has necessarily stabilized to within $\delta_{m,i} \equiv \hat{\beta}_{m+1}|e_m^T q_i^{(m)}|$. If $\mu_i^{(m)}$ is a separated eigenvalue of $\hat{T}_m$ so that (5.26) holds, then it follows from (5.16), (5.17), and (5.28) that the eigenpair $(\mu_i^{(m)}, \hat{V}_m q_i^{(m)})$ represents an eigenpair of A to within

$$3\big(\delta_{m,i} + \sqrt{m}\sigma\epsilon_1\big). \tag{5.33}$$

On the other hand, if $\mu_i^{(m)}$ is one of a close group of eigenvalues of $\hat{T}_m$, so that (5.26) does not hold, then we have found a good approximation to an eigenvalue of A. In this case either $(\mu_i^{(m)}, \hat{V}_m q_i^{(m)})$ represents an eigenpair of A to within (5.33) (see [36]), or there exists $1 \leq r \leq t < m$ such that

$$\max\left(\delta_{t,r}, |\mu_i^{(m)} - \mu_r^{(t)}|\right) \leq \sqrt{3}m^2\sigma\epsilon_2,$$

as from Lemma 5.1. Then, it follows from Theorem 5.2 that $\mu_i^{(m)}$ is within $\big((m+1)^3 + \sqrt{3}m^2\big)\sigma\epsilon_2$ of an eigenvalue of A. The $\delta_{m,i}$ and $\mu_i^{(m)}$ can be computed from $\hat{T}_m$ efficiently, and these results show how we can obtain intervals from them which are known to contain eigenvalues of A, whether $\delta_{m,i}$ is large or small.

6. Convergence of eigenvalues. Theorem 5.2 showed that, assuming (5.1) holds, if an eigenvalue of $\hat{T}_m$ has stabilized to within $\sqrt{3}m^2\sigma\epsilon_2$, then it is within $(m+1)^3\sigma\epsilon_2$ of an eigenvalue of A, regardless of how many other eigenvalues of $\hat{T}_m$ are close, and Corollary 5.3 showed we had an eigenpair of a matrix within $6m^2\sigma\epsilon_2$ of A. It is now shown that, assuming (5.1), eigenvalues do stabilize to this accuracy using the s-step Lanczos method, and we can specify how quickly this occurs.

In [33], it was shown that at least one eigenvalue of $\hat{T}_m$ must have stabilized by iteration $m = n$. This is based on (5.15), which indicates that significant loss of orthogonality implies stabilization of at least one eigenvalue. Using (5.14) and (5.20), if at step $m \leq n$

$$\delta_{\ell,i} \equiv \hat{\beta}_{\ell+1}|\eta_{\ell,i}^{(\ell)}| \geq \sqrt{3}m^2\sigma\epsilon_2, \quad \text{where } 1 \leq i \leq \ell < m, \tag{6.1}$$

then we have

$$\|R_m\|_F^2 \leq \frac{1}{3m^4}\sum_{t=1}^{m-1} t^3 \leq \frac{1}{12}.$$

Let $\sigma_1 \geq \cdots \geq \sigma_m$ be the singular values of $\hat{V}_m$. A result of Rump [41, Lemma 2.2] states that given a matrix $X \in \mathbb{R}^{n\times m}$, if $\|I - X^T X\|_2 \leq \alpha < 1$, then $\sqrt{1-\alpha} \leq$

$\sigma_i(X) \leq \sqrt{1+\alpha}$ for $i \in \{1,\ldots,m\}$. Using the above bound with (4.8), (4.11), and (5.1), we have $\|I - \hat{V}_m^T \hat{V}_m\|_2 < 1$, and then with $\alpha = 2/\sqrt{12} + 1/24$, we apply Lemma 2.2 from [41] to obtain the bounds

$$0.61 < \sigma_i(\hat{V}_m) < 1.3,^2 \quad \text{for} \quad i \in \{1,\ldots,m\}. \tag{6.2}$$

Note that if (6.1) does not hold, then we already have an eigenpair of a matrix close to A. If we now consider the $q_i^{(m)}$ that minimizes $\delta_{m,i}$ for $\hat{T}_m$, we see from (5.14), (5.15), and (5.20) that

$$\|\hat{\beta}_{m+1}\eta_{m,i}^{(m)}\hat{V}_m^T\hat{v}_{m+1}\| \leq m^{3/2}\sigma\epsilon_2. \tag{6.3}$$

THEOREM 6.1. *For the s-step Lanczos method, if $n(3\epsilon_0+\epsilon_1) \leq 1$ and $\epsilon_0 < 1/12$, then at least one eigenvalue of $\hat{T}_n$ must be within $(n+1)^3\sigma\epsilon_2$ of an eigenvalue of the $n \times n$ matrix A, and there exist $r \leq t \leq n$ such that $(\mu_r^{(t)}, z_r^{(t)})$ is an exact eigenpair of a matrix within $6t^2\sigma\epsilon_2$ of A.*

Proof. See [36, Theorem 4.1] and [3, Theorem 6.1]. ∎

This shows that the s-step Lanczos algorithm gives at least one eigenvalue of A to high accuracy by iteration $m = n$, assuming restrictions on the sizes of ϵ_0 and ϵ_1.

We now extend Paige's results to specify how quickly we can expect to find eigenvalues and eigenvectors of A using the s-step Lanczos method in practice. We first consider the Krylov sequence on which the Lanczos algorithm and several other methods are based. For symmetric A, one way of using m steps of the Krylov sequence is to form an $n \times m$ matrix V whose columns span the range of

$$[v_1, Av_1, \ldots, A^{m-1}v_1] \tag{6.4}$$

and use the eigenvalues of

$$V^TAVq = \mu V^TVq \tag{6.5}$$

as approximations to some of the eigenvalues of A. The Lanczos algorithm with full reorthogonalization forms Krylov subspaces for a matrix very close to A with the eigenvalues of T being very close to those of (6.5) [36]. We now show how s-step Lanczos without full reorthogonalization parallels these results.

THEOREM 6.2. *For m iterations of the s-step Lanczos method with (4.10), (5.4), and m such that (6.1) holds, the m Lanczos vectors (columns of $\hat{V}_m$) span a Krylov subspace of a matrix within $(3m)^{1/2}\sigma\epsilon_2$ of A.*

Proof. See [36, Theorem 4.2]. ∎

Comments. This is analogous to the result of Paige for classical Lanczos: until an eigenvalue of $\hat{T}_{m-1}$ has stabilized, i.e., while (6.1) holds, the vectors $\hat{v}_1, \ldots, \hat{v}_{m+1}$ computed correspond to an exact Krylov sequence for the matrix $A+\delta A_m$. As a result of this, and since it follows from (5.16) that $(A + \delta A_r^{(t)})z_r^{(t)} = \mu_r^{(t)} z_r^{(t)}$, if we assume that the s-step bases generated in each outer loop are conditioned such that (5.1) holds, then the s-step Lanczos algorithm can be thought of as a numerically stable way of computing a Krylov sequence, at least until the corresponding Krylov subspace contains an exact eigenvector of a matrix within $6m^2\sigma\epsilon_2$ of A.

[2] Again, these bounds differ from those given in [36], which are $0.41 < \sigma_i(\hat{V}_m) < 1.6$; the present authors suspect that the bounds in [36] were obtained by squaring both sides of the bound and using $\epsilon_0 < 1/100$ rather than $\epsilon_0 < 1/12$, the former being the value used in [33].

When $\hat{T}_m$ and $\hat{V}_m$ are used to solve the eigenproblem of A, if we follow (6.4) and (6.5), we want the eigenvalues and eigenvectors of $\hat{T}_m$ to be close to those of

$$\hat{V}_m^T A \hat{V}_m q = \mu \hat{V}_m^T \hat{V}_m q, \quad \text{where } q^T q = 1, \tag{6.6}$$

as would be the case with classical Lanczos with full reorthogonalization. If (6.1) holds, then the range of $\hat{V}_m$ is close to what we expect from the Lanczos method with full reorthogonalization, and thus the eigenvalues of (6.6) would be close (how close depends on the value of ϵ_2) to those obtained using full reorthogonalization.

THEOREM 6.3. *If $\hat{V}_m$ comes from the s-step Lanczos method with* (4.10) *and* (5.4)*, and* (6.1) *holds, then for any μ and q which satisfy* (6.6)*, $(\mu, \hat{V}_m q)$ is an exact eigenpair for a matrix within $\left(2\delta + 2m^{1/2}\sigma\epsilon_2\right)$ of A, where*

$$\eta \equiv e_m^T q, \quad \delta \equiv \hat{\beta}_{m+1}|\eta|.$$

Proof. See [36, Theorem 4.3] and [3, Theorem 6.3]. ∎

Since from (6.6), $\hat{V}_m^T r = 0$,

$$(\hat{T}_m - \mu I)q = -(\hat{V}_m^T \hat{V}_m)^{-1} \hat{V}_m^T (\hat{\beta}_{m+1}\eta \hat{v}_{m+1} + \delta \hat{V}_m q). \tag{6.7}$$

Then ordering the eigenvalues of $\hat{T}_m$ such that $\delta_{m,1} \geq \delta_{m,2} \geq \cdots \geq \delta_{m,m}$, and assuming (6.1) holds for $\ell = m$, for any eigenpair of (6.6), (6.7) gives, using (4.6), (5.4), (6.2), and (6.3),

$$\|\hat{T}_m q - \mu q\| \leq \left(2 + \frac{3m\delta}{\delta_{m,m}}\right) m^{1/2}\sigma\epsilon_2. \tag{6.8}$$

From this we can write

$$\left(2 + \frac{3m\delta}{\delta_{m,m}}\right) m^{1/2}\sigma\epsilon_2 \geq \|\hat{T}_m q - \mu q\|_2 \geq \min_i |\mu_i^{(m)} - \mu|.$$

Then, from (6.1), $\delta_{m,m} \geq \sqrt{3}m^2\sigma\epsilon_2$, and thus

$$|\mu_x^{(m)} - \mu| \equiv \min_i |\mu_i^{(m)} - \mu| \leq 2m^{1/2}\sigma\epsilon_2 + \frac{\sqrt{3}\delta}{\sqrt{m}}. \tag{6.9}$$

Then, for any $t > m$,

$$\hat{T}_t \begin{bmatrix} q \\ 0_{t-m,1} \end{bmatrix} = \begin{bmatrix} \hat{T}_m q \\ \hat{\beta}_{m+1}\eta e_1 \end{bmatrix},$$

and together with (6.8),

$$\min_i |\mu_i^{(t)} - \mu| \leq 2m^{1/2}\sigma\epsilon_2 + \delta\left(1 + \frac{3}{m}\right)^{1/2}. \tag{6.10}$$

Equations (6.9) and (6.10) can then be combined to give

$$\min_i |\mu_i^{(t)} - \mu_x^{(m)}| \leq 4m^{1/2}\sigma\epsilon_2 + 4\delta.$$

Thus, assuming ϵ_2 is small enough, an eigenvalue of $\hat{T}_m$ close to μ has stabilized to about 4δ, where μ is within 2δ of an eigenvalue of A (see [36, Theorem 4.3]).

It can also be shown that for each $\mu_i^{(m)}$ of $\hat{T}_m$,

$$\min_{\mu \text{ in } (6.6)} |\mu - \mu_i^{(m)}| \leq 2m^{1/2}\sigma\epsilon_2 + \frac{\sqrt{3}\delta_{m,i}}{\sqrt{m}}.$$

This means that when $(\mu_i^{(m)}, \hat{V}_m q_i^{(m)})$ represents an eigenpair of A to within about $\delta_{m,i}$, there is a μ of (6.6) within about $\delta_{m,i}$ of $\mu_i^{(m)}$, assuming $m \geq 3$.

Thus, assuming no breakdown occurs and the size of $\bar{\Gamma}_k$ satisfies (5.1), these results say the same thing for the s-step Lanczos case as in the classical Lanczos case: *until an eigenvalue has stabilized, the s-step Lanczos algorithm behaves very much like the error-free Lanczos process or the Lanczos algorithm with reorthogonalization.*

7. Future work. In this paper, we have presented a complete rounding error analysis of the s-step Lanczos method. The derived bounds are analogous to those of Paige for classical Lanczos [35] but also depend on a amplification factor $\bar{\Gamma}_k^2$, which depends on the condition number of the computed s-step Krylov bases.

We have further shown that the results of Paige for classical Lanczos [36] also apply to the s-step Lanczos method as long as the computed s-step bases remain well-conditioned. As in the classical Lanczos case, the upper bounds in this paper and in [5] are likely large overestimates. We stress, as did Paige, that the value of these bounds is in the *insight* they give rather than their tightness. In practice, the present authors have observed that accurate eigenvalue estimates of A can be found with much looser restrictions than indicated by (5.1), and in some cases even in spite of a numerically rank-deficient basis.

Our analysis and extension of Paige's results confirms the empirical observation that the conditioning of the Krylov bases plays a large role in determining finite precision behavior and also indicates that the s-step method can be made suitable for practical use in many cases, offering both speed and accuracy. The next step is to extend the subsequent analyses of Paige, in which a type of augmented backward stability for the classical Lanczos method is proved [37].

Another area of interest is the development of practical techniques for improving s-step Lanczos based on our results. This could include strategies for reorthogonalizing the Lanczos vectors, (re)orthogonalizing the generated Krylov basis vectors, or controlling the basis conditioning such that (5.1) holds. The bounds could also guide the use of extended or mixed precision in s-step Krylov methods, that is, rather than control the conditioning of the computed s-step base, (5.1) could be satisfied by decreasing the unit roundoff ϵ using techniques either in hardware or software.

REFERENCES

[1] Z. Bai, D. Hu, and L. Reichel, *A Newton basis GMRES implementation*, IMA J. Numer. Anal., 14 (1994), pp. 563–581.

[2] G. Ballard, E. Carson, J. Demmel, M. Hoemmen, N. Knight, and O. Schwartz, *Communication lower bounds and optimal algorithms for numerical linear algebra*, Acta Numer., 23 (2014), pp. 1–155.

[3] E. Carson and J. Demmel, *Accuracy of the s-Step Lanczos Method for the Symmetric Eigenproblem*, Technical report UCB/EECS-2014-165, EECS Department, University of California, Berkeley, CA, 2014.

[4] E. Carson and J. Demmel, *Analysis of the Finite Precision s-Step Biconjugate Gradient Method*, Technical report UCB/EECS-2014-18, EECS Department, University of California, Berkeley, CA, 2014.

[5] E. Carson and J. Demmel, *Error Analysis of the s-Step Lanczos Method in Finite Precision*, Technical report UCB/EECS-2014-55, EECS Department, University of California, Berkeley, CA, 2014.
[6] E. Carson and J. Demmel, *A residual replacement strategy for improving the maximum attainable accuracy of s-step Krylov subspace methods*, SIAM J. Matrix Anal. Appl., 35 (2014), pp. 22–43.
[7] E. Carson, N. Knight, and J. Demmel, *Avoiding communication in nonsymmetric Lanczos-based Krylov subspace methods*, SIAM J. Sci. Comput., 35 (2013), pp. 542–561.
[8] A. Chronopoulos and C. Gear, *On the efficient implementation of preconditioned s-step conjugate gradient methods on multiprocessors with memory hierarchy*, Parallel Comput., 11 (1989), pp. 37–53.
[9] A. Chronopoulos and C. Gear, *s-step iterative methods for symmetric linear systems*, J. Comput. Appl. Math, 25 (1989), pp. 153–168.
[10] A. Chronopoulos and C. Swanson, *Parallel iterative s-step methods for unsymmetric linear systems*, Parallel Comput., 22 (1996), pp. 623–641.
[11] J. Cullum and W. Donath, *A block Lanczos algorithm for computing the q algebraically largest eigenvalues and a corresponding eigenspace of large, sparse, real symmetric matrices*, in Proceedings of the 1974 IEEE Conference on Decision and Control, IEEE, 1974, pp. 505–509.
[12] E. de Sturler, *A performance model for Krylov subspace methods on mesh-based parallel computers*, Parallel Comput., 22 (1996), pp. 57–74.
[13] J. Demmel, M. Hoemmen, M. Mohiyuddin, and K. Yelick, *Avoiding Communication in Computing Krylov Subspaces*, Technical report UCB/EECS-2007-123, EECS Department, University of California, Berkeley, CA, 2007.
[14] J. Demmel, M. Hoemmen, M. Mohiyuddin, and K. Yelick, *Avoiding communication in sparse matrix computations*, in Proceedings of the International Symposium on Parallel and Distributed Processing, IEEE, 2008, pp. 1–12.
[15] D. Gannon and J. Van Rosendale, *On the impact of communication complexity on the design of parallel numerical algorithms*, Trans. Comput., 100 (1984), pp. 1180–1194.
[16] G. Golub and C. Van Loan, *Matrix Computations*, 3rd ed., Johns Hopkins University Press, Baltimore, MD, 1996.
[17] G. Golub and C. Van Loan, *Matrix Computations*, Vol. 3, Johns Hopkins University Press, Baltimore, MD, 2012.
[18] A. Greenbaum, *Behavior of slightly perturbed Lanczos and conjugate-gradient recurrences*, Linear Algebra Appl., 113 (1989), pp. 7–63.
[19] 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.
[20] M. Gustafsson, J. Demmel, and S. Holmgren, *Numerical evaluation of the communication-avoiding Lanczos algorithm*, Technical report ISSN 1404-3203/2012-001, Department of Information Technology, Uppsala University, Uppsala, Sweden, 2012.
[21] M. Gutknecht, *Lanczos-type solvers for nonsymmetric linear systems of equations*, Acta Numer., 6 (1997), pp. 271–398.
[22] A. Hindmarsh and H. Walker, *Note on a Householder implementation of the GMRES method*, Technical report UCID-20899, Lawrence Livermore National Laboratory, Livermore, CA, 1986.
[23] M. Hoemmen, *Communication-avoiding Krylov Subspace Methods*, Ph.D. thesis, EECS Department, University of California, Berkeley, CA, 2010.
[24] W. Joubert and G. Carey, *Parallelizable restarted iterative methods for nonsymmetric linear systems. Part* I: *Theory*, Int. J. Comput. Math., 44 (1992), pp. 243–267.
[25] W. Karush, *An iterative method for finding characteristic vectors of a symmetric matrix*, Pacific J. Math, 1 (1951), pp. 233–248.
[26] S. Kim and A. Chronopoulos, *A class of Lanczos-like algorithms implemented on parallel computers*, Parallel Comput., 17 (1991), pp. 763–778.
[27] S. Kim and A. Chronopoulos, *An efficient nonsymmetric Lanczos method on parallel vector computers*, J. Comput. Appl. Math., 42 (1992), pp. 357–374.
[28] C. Lanczos, *An iteration method for the solution of the eigenvalue problem of linear differential and integral operators*, J. Res. Natn. Bur. Stand., 45 (1950), pp. 255–282.
[29] G. Meurant, *The Lanczos and Conjugate Gradient Algorithms: From Theory to Finite Precision Computations*, SIAM, Philadelphia, 2006.
[30] G. Meurant and Z. Strakoš, *The Lanczos and conjugate gradient algorithms in finite precision arithmetic*, Acta Numer., 15 (2006), pp. 471–542.
[31] M. Mohiyuddin, *Tuning Hardware and Software for Multiprocessors*, Ph.D. thesis, EECS Department, University of California, Berkeley, CA, 2012.

[32] M. Mohiyuddin, M. Hoemmen, J. Demmel, and K. Yelick, *Minimizing communication in sparse matrix solvers*, in Proceedings of the Conference on High Performance Computing Networking, Storage and Analysis, ACM, New York, 2009.
[33] C. Paige, *The Computation of Eigenvalues and Eigenvectors of Very Large Sparse Matrices*, Ph.D. thesis, London University, London, 1971.
[34] C. Paige, *Computational variants of the Lanczos method for the eigenproblem*, IMA J. Appl. Math., 10 (1972), pp. 373–381.
[35] C. Paige, *Error analysis of the Lanczos algorithm for tridiagonalizing a symmetric matrix*, IMA J. Appl. Math., 18 (1976), pp. 341–349.
[36] C. Paige, *Accuracy and effectiveness of the Lanczos algorithm for the symmetric eigenproblem*, Linear Algebra Appl., 34 (1980), pp. 235–258.
[37] C. Paige, *An augmented stability result for the Lanczos Hermitian matrix tridiagonalization process*, SIAM J. Matrix Anal. Appl., 31 (2010), pp. 2347–2359.
[38] C. Paige, I. Panayotov, and J.-P. Zemke, *An augmented analysis of the perturbed two-sided Lanczos tridiagonalization process*, Linear Algebra Appl., 447 (2014), pp. 119–132.
[39] B. Parlett and D. Scott, *The Lanczos algorithm with selective orthogonalization*, Math. Comp., 33 (1979), pp. 217–238.
[40] B. Philippe and L. Reichel, *On the generation of Krylov subspace bases*, Appl. Numer. Math., 62 (2012), pp. 1171–1186.
[41] S. Rump, *Verified bounds for singular values, in particular for the spectral norm of a matrix and its inverse*, BIT, 51 (2011), pp. 367–384.
[42] Y. Saad, *Iterative Methods for Sparse Linear Systems*, SIAM, Philadelphia, 2003.
[43] H. Simon, *The Lanczos algorithm with partial reorthogonalization*, Math. Comp., 42 (1984), pp. 115–142.
[44] R. Thompson and P. McEnteggert, *Principal submatrices* II: *The upper and lower quadratic inequalities*, Linear Algebra Appl., 1 (1968), pp. 211–243.
[45] S. Toledo, *Quantitative Performance Modeling of Scientific Computations and Creating Locality in Numerical Algorithms*, Ph.D. thesis, MIT, Cambridge, MA, 1995.
[46] H. Van der Vorst and Q. Ye, *Residual replacement strategies for Krylov subspace iterative methods for the convergence of true residuals*, SIAM J. Sci. Comput., 22 (1999), pp. 835–852.
[47] J. Van Rosendale, *Minimizing Inner Product Data Dependencies in Conjugate Gradient Iteration*, Technical report 172178, ICASE-NASA, 1983.
[48] H. Walker, *Implementation of the GMRES method using Householder transformations*, SIAM J. Sci. Stat. Comput., 9 (1988), pp. 152–163.
[49] J. Wilkinson, *The Algebraic Eigenvalue Problem*, Vol. 87, Oxford University Press, Oxford, 1965.
[50] S. Williams, M. Lijewski, A. Almgren, B. Van Straalen, E. Carson, N. Knight, and J. Demmel, *s-step Krylov subspace methods as bottom solvers for geometric multigrid*, in Proceedings of the International Symposium on Parallel and Distributed Processing, IEEE, 2014.
[51] W. Wülling, *On stabilization and convergence of clustered Ritz values in the Lanczos method*, SIAM J. Matrix Anal. Appl., 27 (2005), pp. 891–908.
[52] J. Zemke, *Krylov Subspace Methods in Finite Precision: A Unified Approach*, Ph.D. thesis, Technische Universität Hamburg-Harburg, Hamburg, Germany, 2003.