ON THE LONG-TERM BEHAVIOR OF THE LANCZOS PROCESS

ALEXANDER ALPEROVICH, ALEX DRUINSKY, AND SIVAN TOLEDO

Abstract. We investigate the long-term behavior of the classical Lanczos process in an attempt to pave the way to an efficient and robust eigensolver that can find all the eigenvalues of large sparse symmetric matrices. We are interested in the convergence of classical Lanczos (i.e., without re-orthogonalization) to the point where there is a cluster of Ritz values around each eigenvalue of the input matrix A. At that point, convergence to all the eigenvalues can be reliably detected if the matrix has no multiple eigenvalues. To ensure that this is the case, we disperse multiple eigenvalues by adding to A a random matrix with a small norm; using high-precision arithmetic, we can perturb the eigenvalues by an amount that does not affect the accuracy of double-precision computed eigenvalues. Our main results are that Lanczos reliably forms clusters around all the eigenvalues of A and that the speed of cluster formation depends on the local density of eigenvalues and on the unit roundoff. The dependence on the unit roundoff allows us to accelerate convergence by using high-precision arithmetic in computations involving the Lanczos iterates. Clusters form around all eigenvalues after roughly twice the number of iterations required for a single Ritz value to converge to each eigenvalue; therefore, waiting for clusters to form is reasonable. Our detailed experiments reveal additional interesting behaviors, some already known (e.g., misconvergence) and some new (slow divergence of Ritz clusters).

Keywords: Lanczos, mixed precision arithmetic, Ritz clusters

1. Introduction

The Lanczos process is an old and well-known eigensolver [10] (see also [5, 11, 16, 17, 20]). It takes as input an n-by-n Hermitian matrix A and produces sequence of matrices $T^{(m)}$ and $Q^{(m)}$ such that

$$AQ^{(m)} = Q^{(m)}T^{(m)} + r^{(m)}e_m^* ,$$

where $Q^{(m)}$ is n-by-m orthonormal matrix, $T^{(m)}$ is an m-by-m tridiagonal matrix, e_m is the last unit vector of dimension m, and $r^{(m)}$ is some n-vector. The sequences $Q^{(m)}$ and $T^{(m)}$ are nested: each iteration of the Lanczos process adds one column to Q and a row and a column to T. The process is a short-recurrence Krylov-subspace iteration; in each iteration, the algorithm multiplies one vector by A and performs a small number of vector operations on vectors of size n.

In exact arithmetic, the residual vector $r^{(m)}$ vanishes after at most k iterations, where k is the number of distinct eigenvalues of A. When $r^{(m)}$ vanishes, $T^{(m)}$ is an orthonormal projection of A onto the column space of Q, and therefore every eigenvalue of $T^{(m)}$ is an eigenvalue of A. For all the starting vectors except for a set of measure 0, $r^{(m)}$ vanishes after exactly k iterations and all the eigenvalues of A appear in $T^{(k)}$.

Date: November 2011.

Practitioners quickly discovered that the behavior of Lanczos in floating-point arithmetic differs significantly from that predicted by the theoretical results. In particular, the columns of Q quickly lose orthogonality, and r never vanishes in practice. Researchers mostly explored two families of techniques for addressing this difficulty. One set of techniques attempts to prevent the loss of orthogonality in Q. This can be done using a full orthogonalization process or using selective orthogonalization and related techniques [14, 6, 19, 18]. The other set of techniques [1, 21] attempts to extract useful spectral information from the process after a relatively small number of iterations; this rarely results in the identification of all the eigenvalues, but it can result in useful approximations to a subset of the eigenvalues that are important in a given application (e.g., the smallest). These families of techniques are not mutually exclusive; many Lanczos codes use both.

However, around 30 years ago a group of researchers explored the use of Lanczos without sophisticated orthogonalization for finding all the eigenvalues of A [2, 4, 13]; we refer to such methods as *classical Lanczos* methods. This line of research was based on a deep numerical analysis of the Lanczos process that eventually showed that in floating point, *the eigenvalues of T eventually approximate all the eigenvalues of A* [3]. (This fact was recognized years before it was actually proved; see, for example, [2]). These researchers produced two Lanczos codes, both in the 1980s. These codes had to address two major problems: how to decide which of the eigenvalues of T are approximate eigenvalues of A (many are not), and how to decide when to terminate; we describe later how they did it. It appears that there has been no progress in classical Lanczos codes since 1985, although there are some reports about the behavior of these methods in practice [8], as well as some numerical analyses (see [11] and the numerous references therein).

The literature on classical Lanczos does not include a detailed characterization of its convergence behavior. Some papers state that all the eigenvalues of A appear in T within cn iterations for some small constant $c > 1$[8]. Others have observed that c grows with n, but very slowly [4].. The classical-Lanczos literature does not contain extensive experimental results, probably due to the limited computational resources that were available when the codes were written in the 1980s.

The aim of this paper is to explore in more details the long-term behaviors of the classical Lanczos process in the real case. In particular, we address the following questions:

- What factors affect the convergence of the classical Lanczos process, when we take convergence to mean identifiable convergence to all the eigenvalues of A? By *identifiable* convergence we mean that the fact that an eigenvalue of T is genuine (is an accurate approximate eigenvalue of A) can be algorithmically and efficiently identified.
- What is the asymptotic number of iterations required for identifiable convergence? Is it $O(n)$, as suggested in some of the literature? We note that once Lanczos performs $\Theta(n^2)$ iterations, it has done more work and has used more memory than dense eigensolvers, so there is little reason to use it. We are interested in cases where convergence occurs much sooner.

- How does the precision of the floating-point arithmetic (the unit roundoff $\epsilon_{\text{machine}}$) affect the convergence of the process? We are particularly interested in effects that might mitigate slow convergence or non-convergence.

Our long-term goal is to develop a robust classical-Lanczos eigensolver. To do that, we need to know how to set thresholds (e.g., for determining convergence of an eigenvalue), when to use high-precision arithmetic, and how to estimate the running time early on. These design decisions depend on the insights presented in this paper.

The rest of this paper is organized as follows. Section 2 explains how Lanczos algorithms decide that they have found all the eigenvalues of a matrix. This discussion is necessary in order to give a precise meaning for the iteration counts that we present later in the paper. Section 3 starts the exploration of the convergence of the Lanczos process, focusing on matrices with fixed inter-eigenvalues gap. As the Lanczos process progresses, more and more Ritz values cluster around each eigenvalue; we explore this clustering behavior in Section 4. We begin the exploration of more complex spectra in Section 5, where we show that tight clusters in an otherwise nicely-spaced spectrum slow down convergence. In Section 6 we show that the slowdown can be mitigated through the use of high-precision arithmetic. Section 7 studies the asymptotic complexity of the Lanczos process, using carefully constructed spectra, using the insights gained in earlier sections. Our conclusions from this study are presented in Section 9.

2. Detecting Termination

A robust Lanczos eigensolver needs to terminate once it found all the eigenvalues. This may seem obvious, but the literature does not describe reliable ways to do that. We believe that a Lanczos eigensolver that reliably finds all the eigenvalues of A and terminates can be designed using a combination of two techniques. To effectively use these techniques, we need to know more about the convergence of the Lanczos process; this paper is an effort to generate this knowledge. This section outlines these two techniques and discusses what we need to know about Lanczos convergence to use them. The actual investigation of the two techniques is beyond the scope of this paper.

The first technique is eigenvalue dispersal. We add a random matrix P with a small norm to the input matrix A in order to transform every multiple eigenvalue of A and every tight eigenvalue cluster into a not-so-tight cluster of simple eigenvalues of $A+P$. We run the classical Lanczos algorithm on $A+P$, which has exactly n simple eigenvalues.

The other technique uses the spectrum of $T^{(m)}$ to reliably locate eigenvalues of $A+P$. Once we find n disjoint intervals that each contain an eigenvalue, we have found all the eigenvalues of A to within an error determined by the size of the interval and the norm of P, and we terminate.

We begin by explaining the dispersion idea.

2.1. **Eigenvalue Dispersal.**

Independently of how the code determines which Ritz values are genuine, it also needs to decide when to stop. The Lanczos process provides no information on the multiplicity of the eigenvalues of A. Therefore, when some number $k < n$ of eigenvalues have been found, they might constitute the entire spectrum of A (if A has multiple eigenvalues), or they might be a proper subset, with some eigenvalues of A still to be found. This problem has plagued all the classical Lanczos codes.

Our code will use a conceptually simple solution that we call *dispersion*. Instead of running Lanczos on A itself, we will run it on $A + P$, where P is a random symmetric matrix (from some appropriate distribution) with a small norm $\|P\|_2 \leq \delta$. We chose P so that it is cheap to apply to vectors; this results in Lanczos iterations that are about as cheap as those performed on A alone. The perturbation P perturbs the eigenvalues, but only by δ or less. Hopefully, $A+P$ has no multiple eigenvalues; multiple eigenvalues of A are transformed into clusters of close but distinct eigenvalues of $A+P$. The choice of P determines how close the eigenvalues of $A + P$ are; we do not have a complete theory that guarantees good separation with high probability, but experiments have shown that dispersion works well. We omit these experiments from this paper, and focus instead on the convergence for a given operator (which the reader can take to be $A + P$).

We note that if the user would like to determine the eigenvalues to within a small tolerance ϵ near $\epsilon_{\text{machine}}$, the norm δ of the perturbation P will need to be even smaller, which implies that multiple eigenvalues of A will be transformed into very tight clusters in $A + P$; these clusters will behave in floating point exactly like multiple eigenvalues.

The solution in this case is to shrink $\epsilon_{\text{machine}}$ by resorting to high-precision arithmetic. This technique essentially widens the gap between the desired accuracy ϵ and the unit roundoff $\epsilon_{\text{machine}}$ so that a convenient value of δ in between them can be chosen.

2.2. **Locating Eigenvalues.** A growing body of results suggest that non-trivial clusters of Ritz values are only found very close to eigenvalues of A. That is, if we find two or more eigenvalues of $T^{(m)}$ that are very close to each other, they normally indicate the location of an eigenvalue of A; we call such Ritz values *doubly-converged*. This phenomenon was known to Cullum and Willoughby [2] and to Parlett and Reid [13], but back then there were no provable bounds on the location of eigenvalues relative to non-trivial Ritz clusters. Furthermore, both groups aimed to detect convergence even before clusters form, so their codes also used more poorly justified eigenvalue-location estimates.

In the years that followed, double convergence was analyzed more rigorously, and we now know that clusters of Ritz values normally indicate the location of eigenvalues. We say *normally* because all the results in the literature are conditioned on properties of the spectrum of A and/or $T^{(m)}$ which might not hold. However, exceptions seem very rate, and some conditions are easily tested (in particular, conditions that only involve Ritz values are easy to test).

In the rest of this section, we describe two such results and we explain what kind of experimental analysis is required to understand them more fully.

We begin with a result of Wülling [22]. Assume that the Lanczos process (in floating-point) for $m > n$ iterations and compute the eigendecomposition of $T^{(m)} = XRX^T$. The matrix X is unitary and its columns are the eigenvectors of $T^{(m)}$, and R is diagonal and its diagonal entries are the eigenvalues of $T^{(m)}$ (Ritz values of A). The eigendecomposition of a tridiagonal symmetric matrix costs $\Theta(m^2)$ arithmetic operations, can be done sequentially within a memory of size $\Theta(m)$ (if X is computed one column at a time), and can be easily parallelized to at least m processors.

It is well known that when $X_{m,j}$ is small, then $R_{j,j}$ is close to an eigenvalue of A [12, p. 249],

$$\min_{1\le k\le n} |\lambda_k - R_{j,j}| \le 2.5 T^{(m)}_{m+1,m} X_{m,j} + \|A\|\, O\left(\epsilon_{\text{machine}}\right) \ .$$

This bound gives us m intervals that contain all the eigenvalues of A. If we find among them n disjoint intervals, we are done; we have located all the eigenvalues.

Will we always find n disjoint intervals and will they always be small enough for some reasonable iteration count m? Wülling [22] showed that each tight cluster of Ritz values that is well separated from all other clusters corresponds to at least one small $X_{m,j}$ (where j is part of the cluster); this implies that the cluster is close to some λ_k. Wülling's result depends on the cluster remaining non-trivial (with 2 or more Ritz values) for two consecutive iterations (say m and $m-1$); he shows an example in which symmetry causes the size of one cluster to oscillate between 1 and 2, but this seems to depend on both symmetry in the spectrum and on a particular choice of the Lanczos starting vector. We have found that floating-point errors quickly destroy the symmetry even in this example, leading to two consecutive iterations in which the cluster has 2 Ritz values.

Wülling's bounding interval depends on both the cluster size, its diameter, and the separation from other Ritz values. A large well separated and tight cluster leads to a small bound on $X_{m,j}$; a small cluster or one that is not well separated or one that is fairly spread out yields a large, perhaps useless bound on $X_{m,j}$. Wülling's bound is much sharper if the cluster size grows or shrinks between iterations m and $m-1$, but we can't expect to find more than one such cluster if we decompose $T^{(m)}$ and $T^{(m+1)}$ for just one value of m.

Knizhnerman [9, Theorem 2], sharpening an earlier result of Greenbaum [7], provides a different kind of guarantee on Ritz clusters. Knizhnerman showed that if there are no Ritz values that are close but not very close to eigenvalues of A, then every Ritz cluster is close to an eigenvalue. More specifically, Knizhnerman's result assumes that there are no Ritz values within distance

$$\Omega(m^3\epsilon_{\text{machine}}\|A\|) \le d \le \max\left(O(m^3\epsilon_{\text{machine}}\|A\|), O(m\epsilon^{1/3}_{\text{machine}}\|A\|)\right)$$

of an eigenvalue of A, where the constants within the big-O and big-Ω notation depend on n (linearly) and on $\||A|\|/\|A\|$.

Our approach in this paper is to simply assume that a doubly-converged Ritz value indicates the location of an eigenvalue; the results of Wülling, Paige, Knizhnerman and Greenbaum suggests that this is usually the case. Once we found n distinct clusters, the code can compute X to ascertain that we have indeed found all the eigenvalues of A. Our experiments aim to demonstrate the validity of this approach by examining the following aspects:

- We show that Ritz values do cluster around eigenvalues and that the diameter of the cluster is a function of $\epsilon_{\text{machine}}$. This implies that codes can use high-precision arithmetic to force clusters to be tight.
- We also show that additional Ritz values cluster around each eigenvalue periodically, with a periodicity that changes from eigenvalue to eigenvalue. This

implies that clusters always form around eigenvalues; perhaps slowly, but they always form.

- Finally, we show that the periodicity of each cluster depends on both the shape of the spectrum and on $\epsilon_{\text{machine}}$. High-precision arithmetic reduces the variability among the periods of the different clusters, reducing the risk of some clusters forming very slowly.

2.3. **Termination Conditions in Early Codes.** The results that we cited above are relatively new; they were not known when the Cullum-Willoughby and Parlett-Reid codes were written. Both codes used heuristics to locate eigenvalues and to terminate. Both codes reported approximate eigenvalues early (perhaps too early). Cullum and Willoughby assumed that when a Ritz value is almost an eigenvalue of a matrix obtained from $T^{(m)}$ by deleting the first row and column, it is close to an eigenvalue of A. They have an argument that supports this criterion but no formal proof. Parlett and Reid keep track of intervals with no Ritz values and at some point declare them as being outside the spectrum of A, also heuristically. Both codes take a cluster of size 2 to indicate an eigenvalue of A. None of the codes attempt to find n eigenvalues because they assume that the matrix might have multiple eigenvalues.

3. Convergence when the Eigenvalues of A are Regularly Spaced

We now begin the exploration of the convergence behavior of Lanczos. In order to explore it systematically we begin with very simple cases and then progress to more complex ones. This strategy allows us to understand individual behaviors in isolation. Long-term behavior: regularly spaced eigenvalues and a regular starting vector. As we'll see later, clustered Eigenvalues have a significant effect on convergence, and so does the starting vector. We begin the exploration with matrices that have no clusters at all: their eigenvalues are regularly spaced in the spectrum. The matrices are diagonal. The starting vector has $1/\sqrt{n}$ in all the entries. Because the eigenvectors are unit vectors, the projection of the starting vector on all the eigenvectors is the same.

Figure 3.1 shows the location of Ritz values over 5000 iterations of the Lanczos algorithm on a matrix of dimension $n = 200$ with regularly spaced eigenvalues. (Figure 3.2 visualizes the same Lanczos process differently.) The horizontal lines fall on eigenvalues. The darker areas in the figure are areas in which Ritz values have not yet converged, so they move from iteration to iteration.

The main artifact that we see is that Ritz values converge first at the outer edges of the spectrum. As the algorithm continues, Ritz values converge to eigenvalues in the interior of the spectrum. When there is a converged Ritz value near every eigenvalue, new Ritz values start to appear at the outer edges again. They do wander a bit and then converge, leading eventually to two Ritz values near every eigenvalues. The process continues.

Ritz values converge faster near the edges of the spectrum; the crescent-shaped areas of non-converged Ritz values are thinner at the edges and thicker in the center of the spectrum.

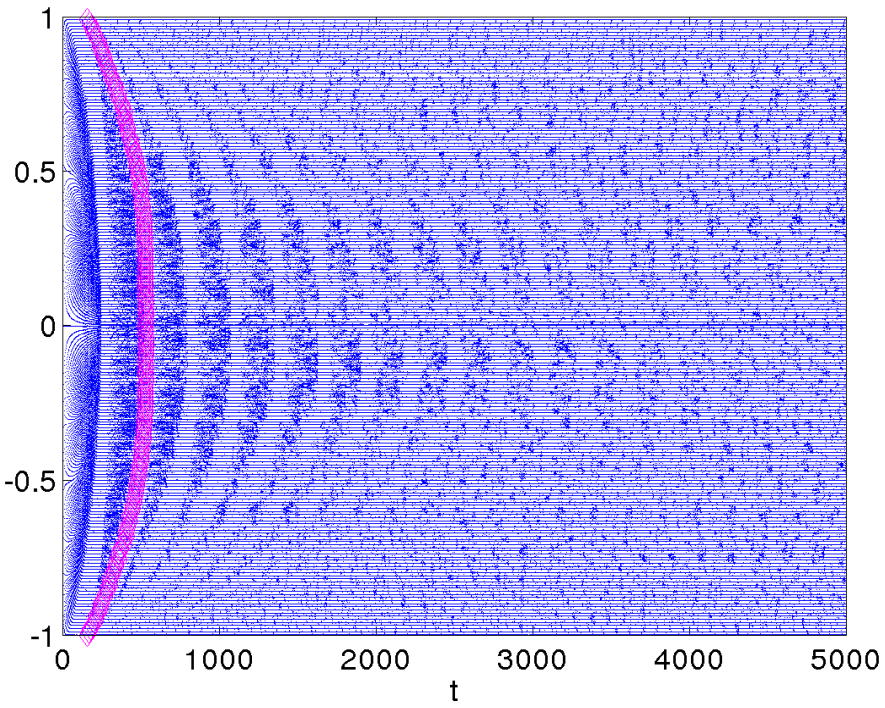

FIGURE 3.1. The evolution of the Ritz values as the number of Lanczos iterations grows. Each blue dot represents an eigenvalue of T. The purple circles show points of double convergence (a Ritz cluster growing to size 2).

Over time, it takes less time for Ritz values to converge. This phenomenon manifests itself on the graph by the disappearance of the crescent-shaped bands of non-converted Ritz values.

A closer inspection of the graph in Figure 3.1 shows that the curvature of the crescents grows. After, say, 1500 iterations, there are more Ritz values near eigenvalues at the edges of the spectrum than near eigenvalues at the center. Figure 3.3 quantifies this more clearly. For a matrix of dimension 200 the ratio is around 3.5 to 3.75 and it is stable as the number of iterations grows.

As the matrix dimension grows, the ratio between the number of converted Ritz values at the edges and converged Ritz values at the center grows. Figure 3.4 shows the numbers for a matrix of dimension $n = 2000$; the ratio is about 5. The ratio remains stable as the number of iterations grows. We also see that the behavior at the center is valid over a larger part of the spectrum than for the smaller matrix in Figure 3.3; here the number of converged Ritz values is fairly flat in most of the spectrum; it rises dramatically only close to the edges of the spectrum.

The effect of the starting vector. When the projection of the starting vector on a particular eigenvector is small, convergence to the corresponding eigenvalue is slower than when the projection is large. However, Figure 3.5 shows that the effect is not necessarily dramatic. When half the projections are large and half are small, convergence

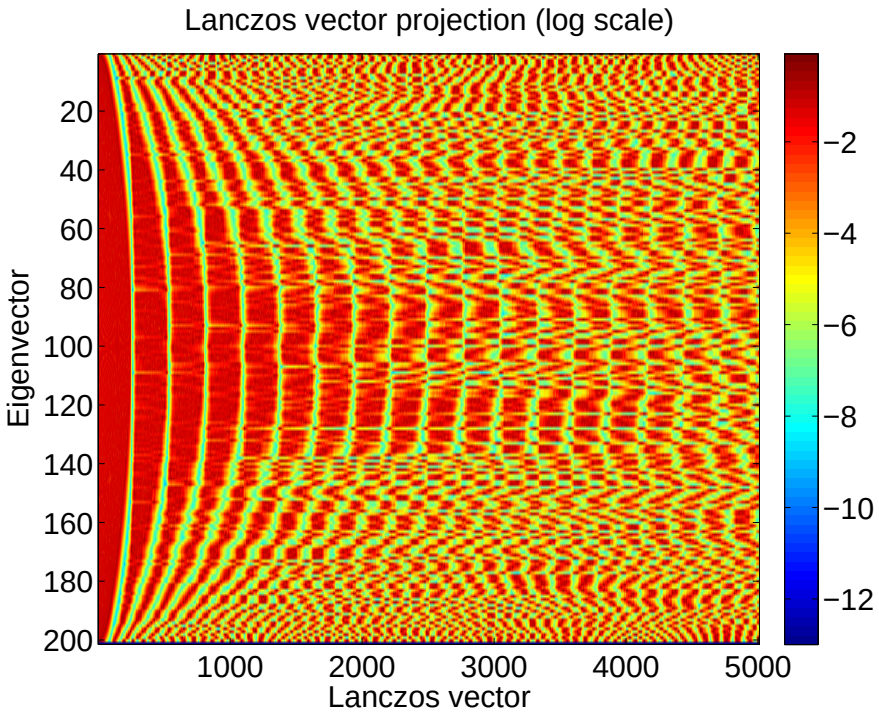

FIGURE 3.2. The projection of the Lanczos basis vectors on the eigenvectors of the matrix (the same matrix as in Figure 3.1). The color scale is logarithmic with base 10.

to the eigenvalues associated with small projections is slower, but not dramatically so. Even when the small projections were smaller by a factor of 10^{16} (they were near the unit roundoff $\epsilon_{\text{machine}}$), a Ritz value converged to each eigenvalue after fewer than $2n$ iterations.

4. The Behavior of Ritz Clusters

As the number of iterations grows, more and more Ritz values converge to each eigenvalue. In general, this is a good thing; two very close Ritz values indicate that they must be close to an eigenvalue.

We discovered two behaviors of these Ritz clusters that were not previously observed. These behaviors are important for the design of Lanczos codes.

The first behavior, shown in Figure 4.1, is a divergence of the clusters. As the number of Ritz values in the cluster grows, the cluster widens. Each widening even grows the cluster only by a small amount, but it does not appear that this process has a limit.

The scale of the divergence is determined by the unit roundoff. Figure 4.2 shows that when the Lanczos code is implemented in double-double precision (128-bit floating-point numbers) the divergence looks similar qualitatively but the scale of the divergence shrinks by about 16 decimal digits.

The divergence of Ritz clusters has two implications for the design of Lanczos codes. The first is that eigenvalues should be extracted from Ritz values as soon as possible.

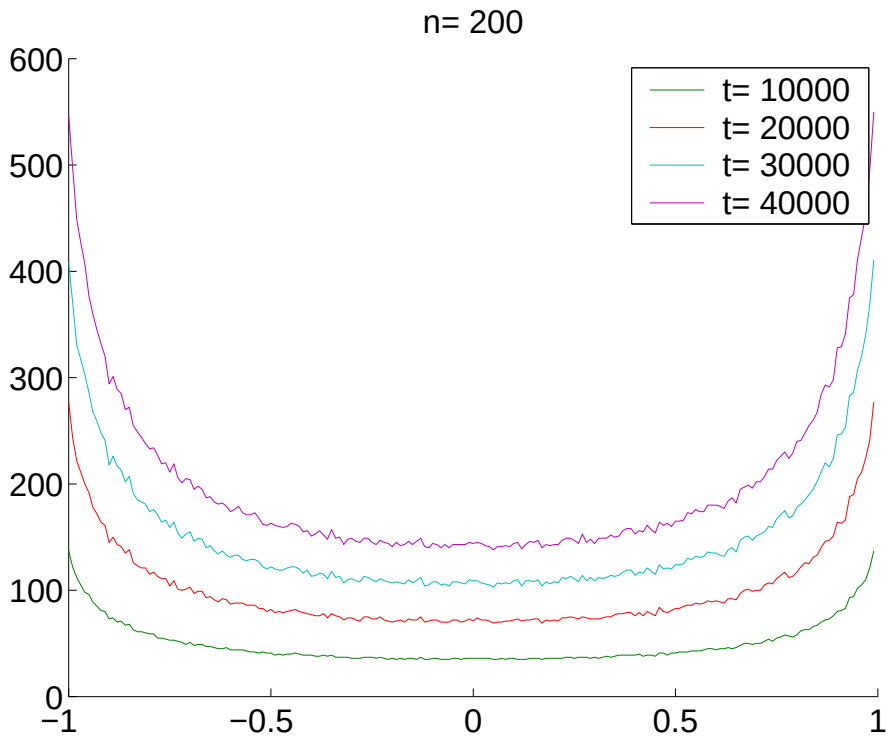

FIGURE 3.3. The number of Ritz values (for $n = 200$) near each eigenvalue after 10,000 to 40,000 iterations. The graph only count converged Ritz values (closer than 10^{-13} to an eigenvalue).

If the iteration continues for a long time because Ritz values failed to converge to some eigenvalue (we will see below that this happens when the eigenvalues themselves are clustered), the code should estimate the other eigenvalues from early Ritz values, not from the Ritz values at the end of the iterations. The other implication is that continuous ranges of Ritz values do not necessarily imply the discovery of new eigenvalues; they may indicate a divergence from one eigenvalue.

The second behavior of clusters is a dependence of the widening on the location in the spectrum. Figure4.3 shows that in the middle of the spectrum, Ritz values in a cluster are spaced much more widely than at the edges of the spectrum. In both cases we see clusters of Ritz values near eigenvalues, but the clusters are tighter at the edges. We also see that when the cluster grows by one Ritz value, old Ritz values are replaced by new ones; Ritz values never seem to be completely converged.

Spectrum with Gaps. Ritz values tend not to fall within large gaps in the spectrum, as shown in Figure 4.4. The density of Ritz values within the gap is much smaller than their density outside the gap, and they do not exhibit any convergence behavior in the gap. No Ritz values fall outside the spectrum, except perhaps for small deviations near the extreme eigenvalues.

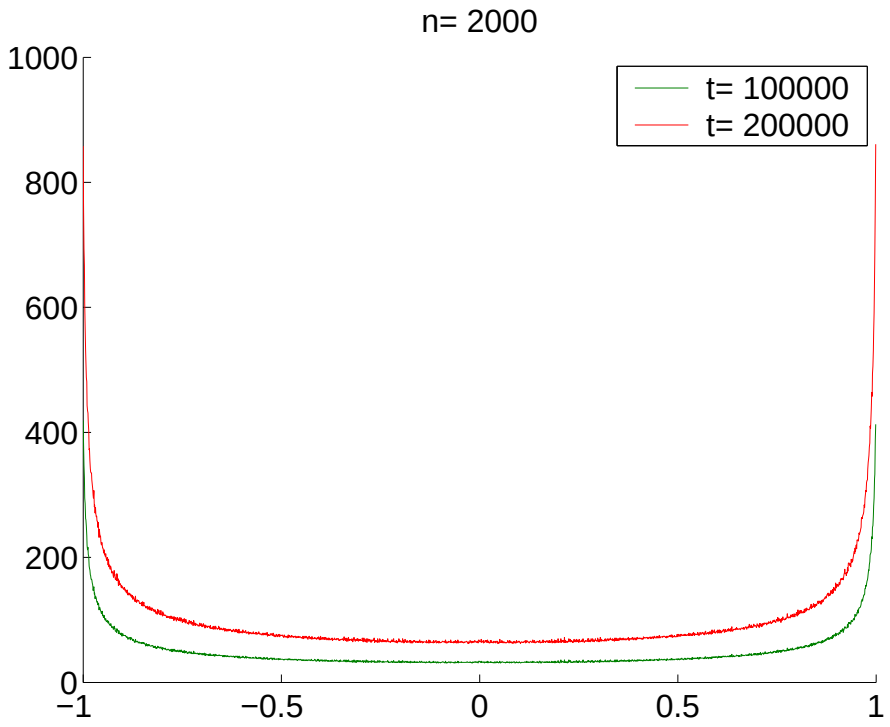

FIGURE 3.4. The number of converged Ritz values for a larger matrix ($n = 2000$) after 100,000 and 200,000 iterations.

On the other hand, even when the gap is large, eigenvalues on the far side of the gap influence convergence. Ritz values converge more quickly at the outer edges of the spectrum than near the inner edges adjacent to the gap.

We conclude that gaps in the spectrum do not appear to slow down convergence; we will see below that this is not true for clusters the in spectrum.

5. The Effects of Clusters of Eigenvalues

Clusters of eigenvalues affect the convergence of Lanczos in ways that are important for the design of Lanczos codes. Parlett [15] showed that a Ritz value can converge to a value between two eigenvalues and hold there for a number of iterations before moving to an eigenvalue. In this section we study this phenomenon, which Parlett calls *misconvergence*, as well as several other cluster-related behaviors.

A small cluster (10 eigenvalues out of 200) does not affect much convergence outside the cluster, as shown in Figure 5.1; the overall behavior is similar to the one in Figure 3.1.

On an eigenvector/Lanczos-vector projection map, the cluster is easily visible, as can be seen in Figure 5.2; the behavior in the cluster is clearly different from the behavior outside the cluster.

Tight clusters cause severe misconvergence. Figure 5.3 shows that a Ritz value that shows up in a cluster tends to wander around near and between eigenvalues and then typically settles for a long time *in-between* eigenvalues. As more Ritz values show

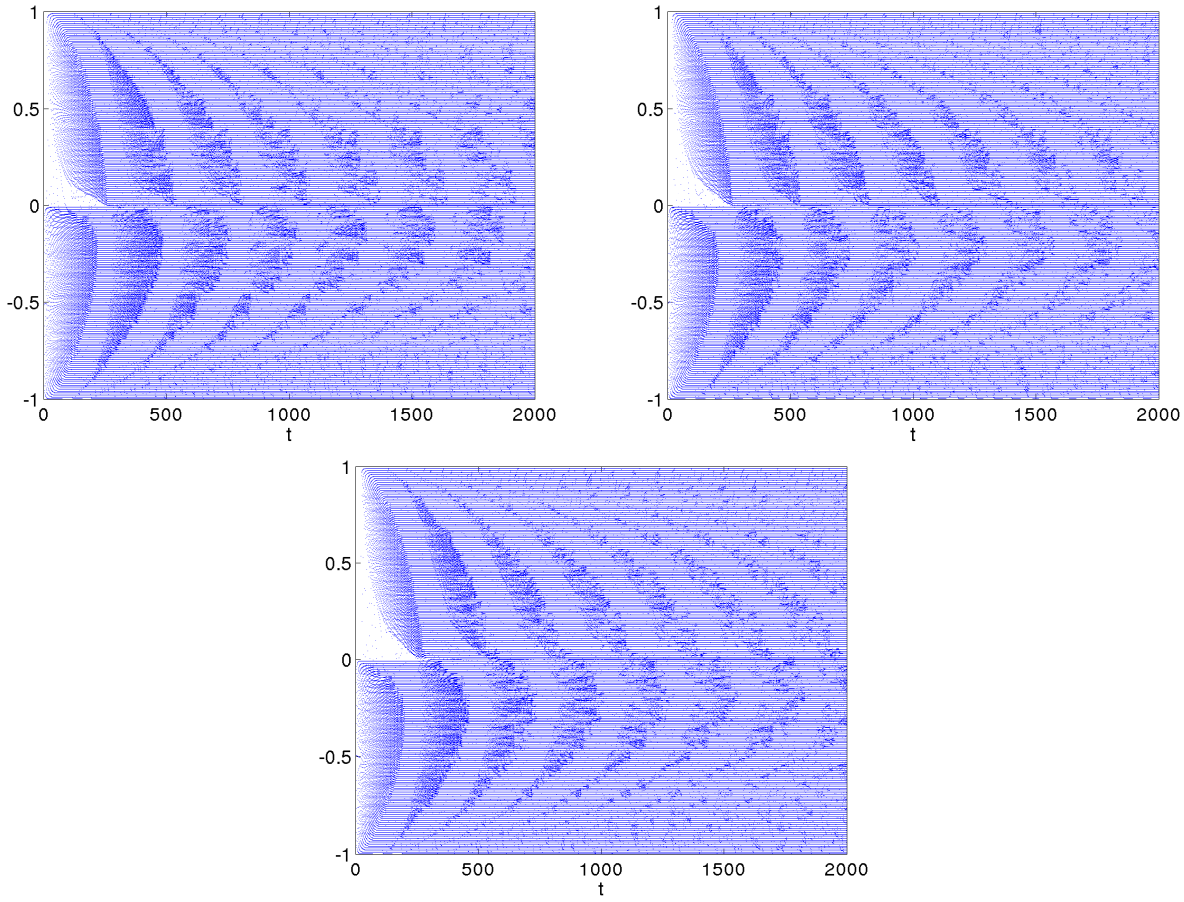

FIGURE 3.5. The evolution of the Ritz values as the number of Lanczos iterations grows. The matrix has dimension $n = 200$ and regularly-spaced eigenvalues. The projection of the starting vector (before normalization) on the smallest $n/2$ eigenvalues is random between 1 and 2 and the projection on the largest eigenvectors is between 1×10^{-k} and 2×10^{-k}, for $k = 8$ (top left), $k = 12$ (top right), and $k = 16$ (bottom).

up, a misconverged eigenvalue tends to shift closer to an eigenvalue, until it actually converges. If we inspect the eigenvalue at 5×10^{-12}, for example (the top most one), we see a misconverged Ritz value that shifts between 3 or 4 stable locations before converging. The bottommost Ritz value in the cluster shows a symmetric behavior. The periods of misconvergence tends to be similar to each other; they are probably dictated by the periodicity of the appearance of new Ritz values in the cluster.

The most important effect of clusters is on the periodicity of the appearance of Ritz values near eigenvalues. In a cluster, the periodicity is longer; a Ritz value appears near a specific eigenvalue less often than near non-clustered eigenvalues. This is shown in the left plot of Figure 5.4. This phenomenon causes Lanczos to converge more slowly to all the eigenvalues when there are clusters than when the the eigenvalues are regularly spaced. If we examine the raw density of Ritz values, ignoring the distribution of

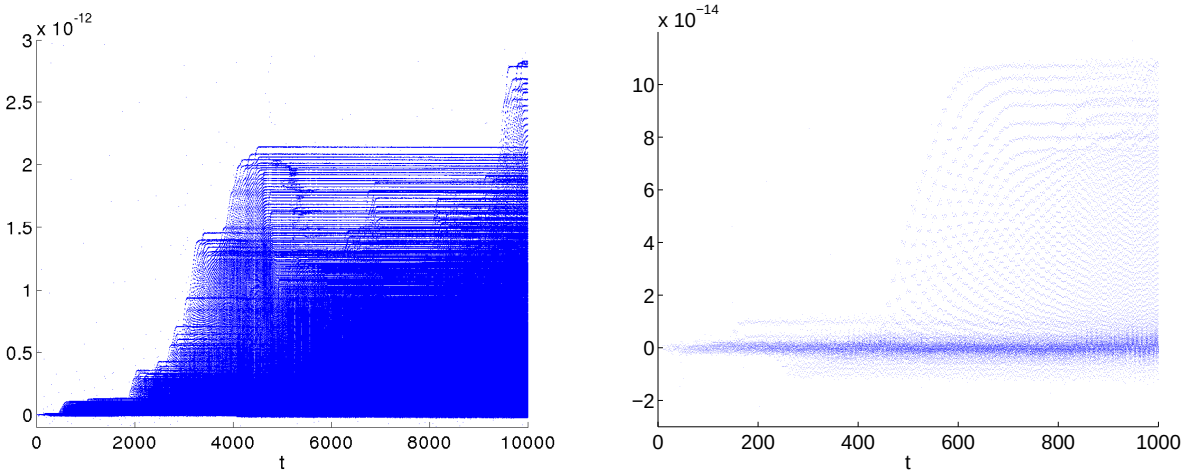

FIGURE 4.1. Ritz values near the eigenvalue 0 in a 10-by-10 matrix. In this experiment the Ritz values in the cluster mostly diverge away from the center of the spectrum (the center is at -0.1), but this depends on the starting vector. The starting vector here is the constant vector. The graph on the left shows the behavior over 10,000 iterations whereas the graph on the right shows only the first 1,000 iterations.

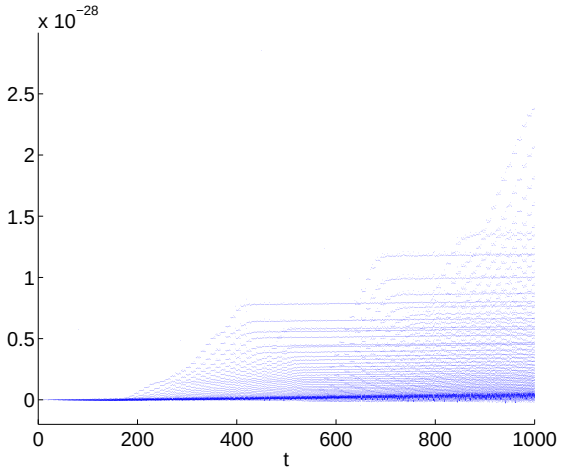

FIGURE 4.2. Divergence using double-double precision (about 32 decimal digits in the fraction of a floating-point number). We still see divergence, but on a much smaller scale.

eigenvalues, we see that the cluster attracts more Ritz values than intervals of the same size elsewhere in the spectrum. This is shown in the right plot of Figure 5.4. This increased attraction is not sufficient, however, to compensate for the larger number of eigenvalues in the interval, so convergence to all eigenvalues is still adversely affected by the cluster.

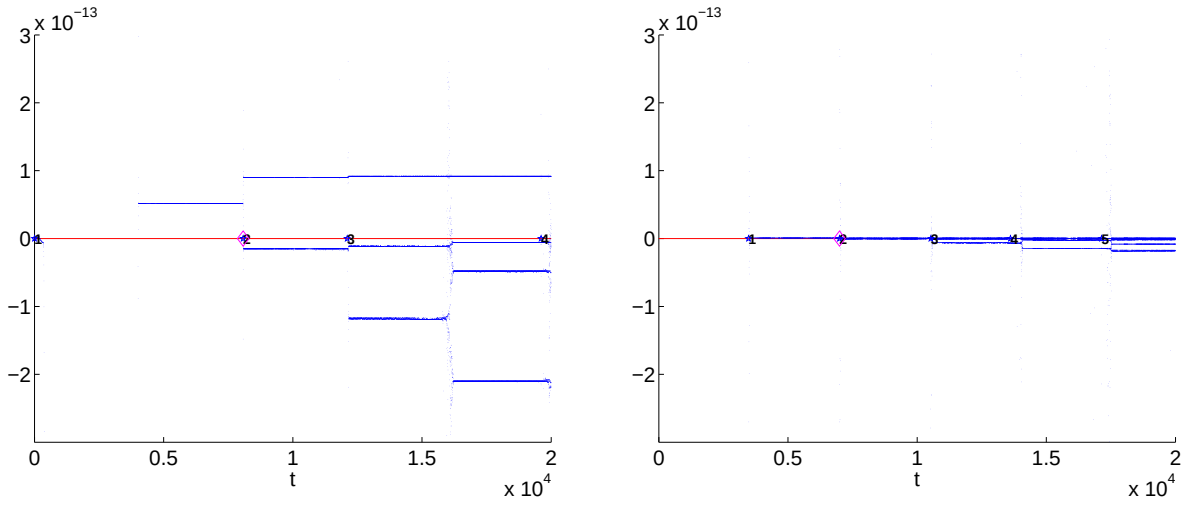

FIGURE 4.3. Divergence of Ritz values in the center of the spectrum (left) and half way from the center to the edge of the spectrum (right). The red line represents the eigenvalue and blue dots represent Ritz values. The labels on the Y axes are relative to the eigenvalue.

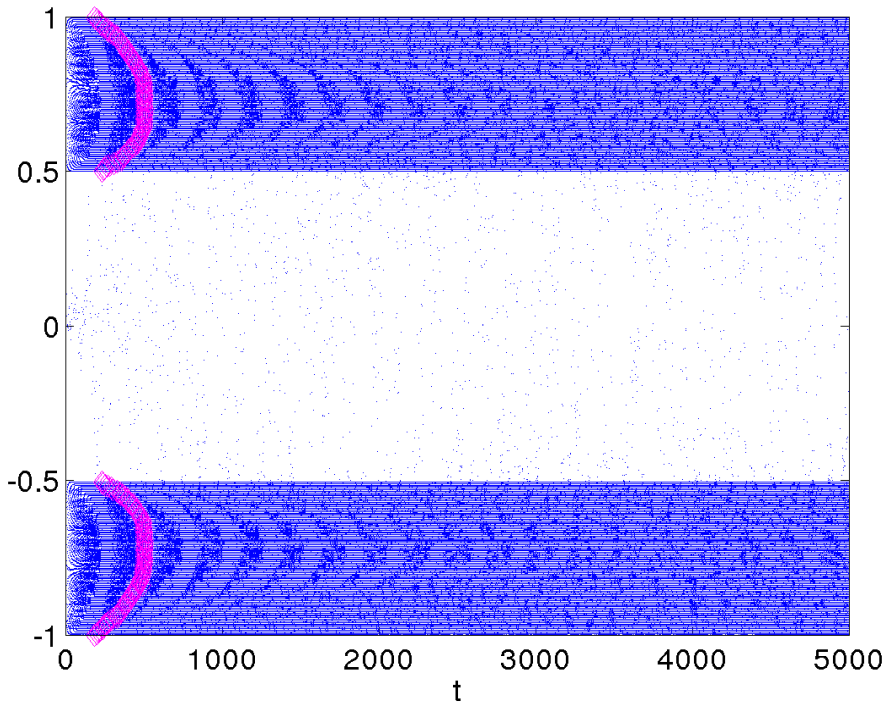

FIGURE 4.4. The evolution of Ritz values when the eigenvalues are regularly spaced in *two* intervals, from $[-1, -0.5]$ and $[0.5, 1]$. The starting vector is random and $n = 200$.

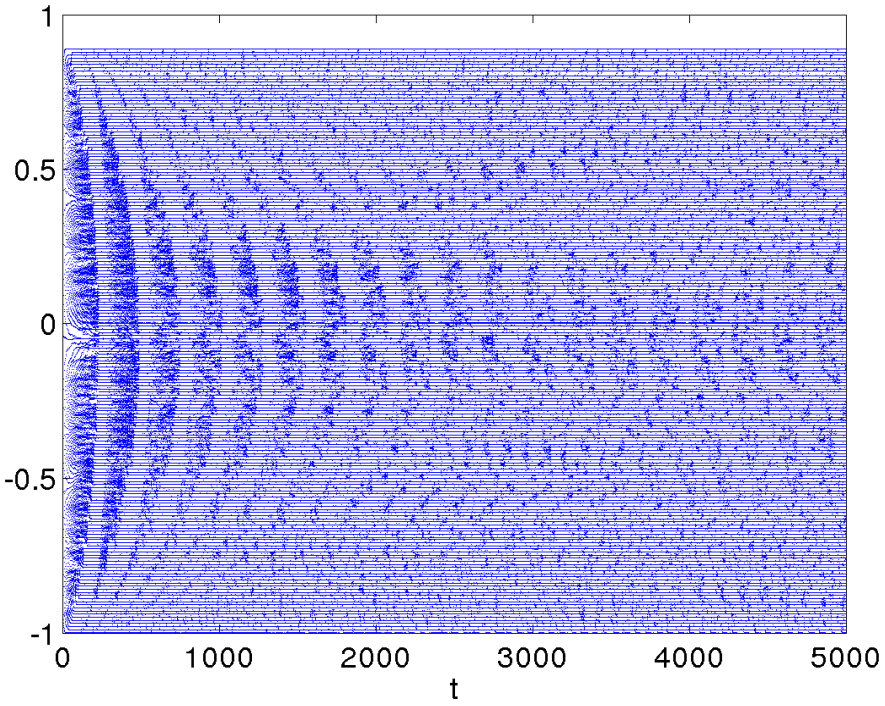

FIGURE 5.1. The evolution of the Ritz values as the number of Lanczos iterations grows for a matrix with a cluster of 10 eigenvalues (spaced 10^{-12} apart) located in the center of the spectrum. Each blue dot represents a Ritz value.

6. The Effects of High-Precision Arithmetic on the Lanczos Process

Increasing the precision of the floating-point arithmetic reduces the adverse effect of clusters, as shown in Figure 6.1. As we increase the precision, the number of Ritz values in a cluster increases, speeding up the convergence. This phenomenon was already observed by Edwards et al. [4], but it does not appear that Lanczos codes used this insight.

Figure 6.2 shows that high precision only helps if the cluster is not too tight. If the cluster is tight relative to $\epsilon_{\text{machine}}$, it attracts too few Ritz values even if $\epsilon_{\text{machine}}$ is small; what matters is the spacing of eigenvalues in the cluster relative to $\epsilon_{\text{machine}}$.

The high precision is important only for the Lanczos iteration vectors, from which the tridiagonal matrix T is constructed. Once computed, T can be rounded to a lower precision (double precision in our experiments) and its eigenvalues computed in that precision without affecting the overall convergence behavior. This is significant since computing the eigenvalues of T is typically more expensive than producing it, at least when A is reasonably sparse.

As the size of an eigenvalue cluster grows, its effect on convergence becomes devastating, even in high precision. Figure 6.4 shows that as the size of a cluster grows, the number of Ritz values in it increases, but not nearly fast enough to obtain convergence on all eigenvalues. When the cluster is small, say containing 10 eigenvalues, there are

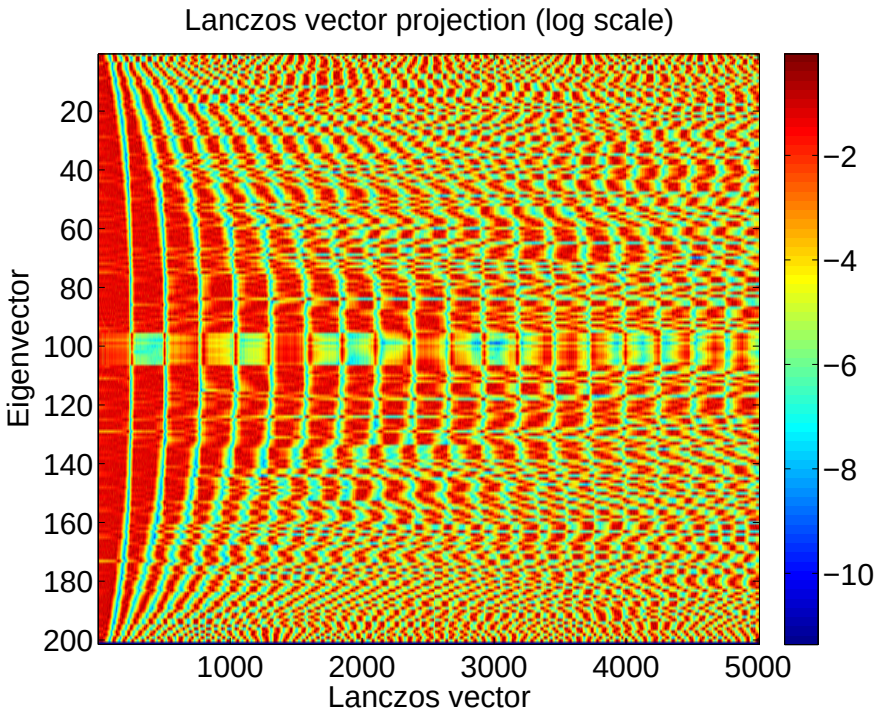

FIGURE 5.2. The projection of the Lanczos basis vectors on the eigenvectors of the matrix (the same matrix as in Figure 5.1). The color scale is logarithmic with base 10.

more than 3 Ritz values per eigenvalue after $20n$ iterations, even in 64-bit arithmetic (and more Ritz values in higher precision). When the cluster contains 100 eigenvalues, there is not even a single Ritz value per eigenvalue after $20n$ iterations; we cannot expect convergence in that many iterations. Things get much worse as the cluster size continues to grow.

The distribution of Ritz values within the cluster is typically *not* uniform, just like within the spectrum as a whole. When eigenvalues in the cluster are distributed uniformly, more Ritz values appear at the edges of the cluster than near its center, as shown in Figure 6.5. This implies that even once there are *on average* 2 or 3 Ritz values per eigenvalue in the cluster, we may be very far from convergence, because there are not enough Ritz values near eigenvalues in the center of the cluster.

7. The asymptotic complexity of the Lanczos process

To assess the asymptotic convergence of the Lanczos process, we ran it on two family of synthetic matrices. In both families, all the eigenvalues belong to two equal-size clusters, one at 1 and the other at -1. In the first family, we kept the inter-eigenvalue separation δ within a cluster fixed, at $\delta = 10^{-8}$. This allows us to determine the effect of growing matrix dimension without causing the eigenvalues to get closer to each other. In the second family, we kept the matrix dimension at $n = 1000$ while shrinking the inter-eigenvalue distance within each cluster.

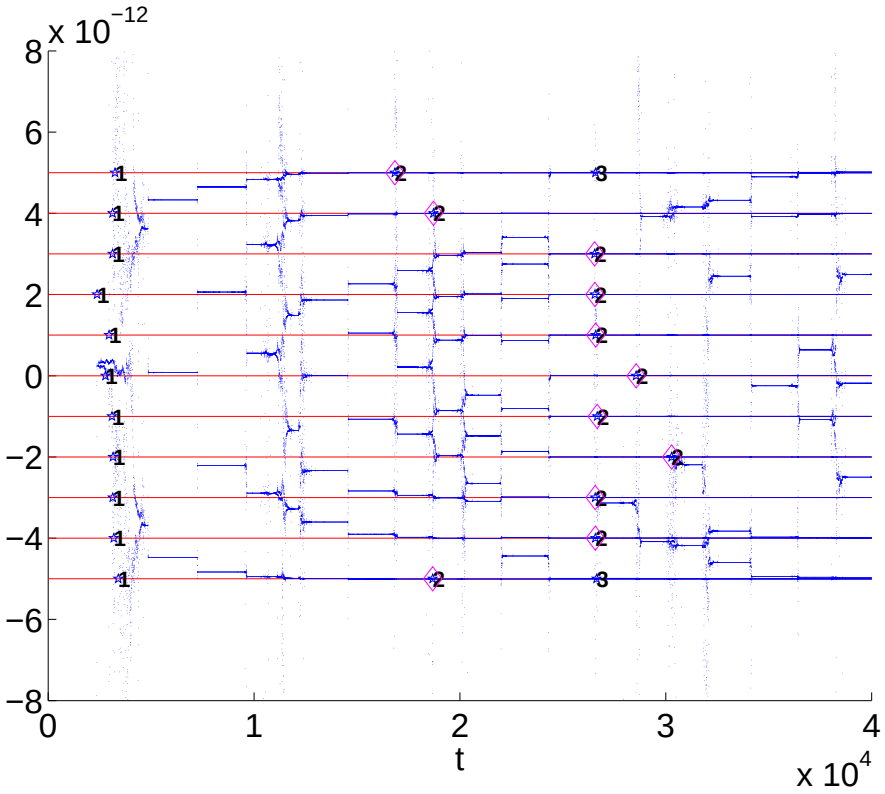

FIGURE 5.3. Evolution of Ritz values near a cluster. The red lines represents a cluster eigenvalues (they are 10^{-12} apart) and blue dots represent Ritz values. The purple circles show where double convergence first occurs. A numeral k shows the first time that there are k Ritz values near an eigenvalue.

The results, shown in Figure 7.1, show that for a given clustering of the eigenvalues, the growth in iteration count is proportional to $c(n) \times n$, where $c(n)$ is a very slowly growing function of n. The inter-eigenvalue separation has a very slight effect on convergence.

The fact that the inter-eigenvalue gap does not influence much the convergence rate seems to contradict the results of Section 5, but it does not. A tight cluster in a spectrum with mostly wide inter-eigenvalue gaps attracts too few Ritz values and slows down convergence. In a spectrum in which all the eigenvalue gaps are small, the Ritz values do appear within the spectrum (but rarely in huge gaps, as shown in Figure 4.4), so convergence is not affected by the magnitude of the gaps. Figure 7.2 demonstrates that this is indeed the case. When the eigenvalues at 1 are spaced 10^{-6} apart but the eigenvalues at -1 were spaced δ apart for $\delta = 10^{-10}, \ldots, 10^{-6}$, convergence depends more strongly on δ.

When the eigenvalues are regularly spaced, larger matrices require more iterations to converge, because their eigenvalues are closer together, as shown in Figure 7.3. Convergence to a fixed tolerance tends to occur after $k \times cn$ iterations for some constant

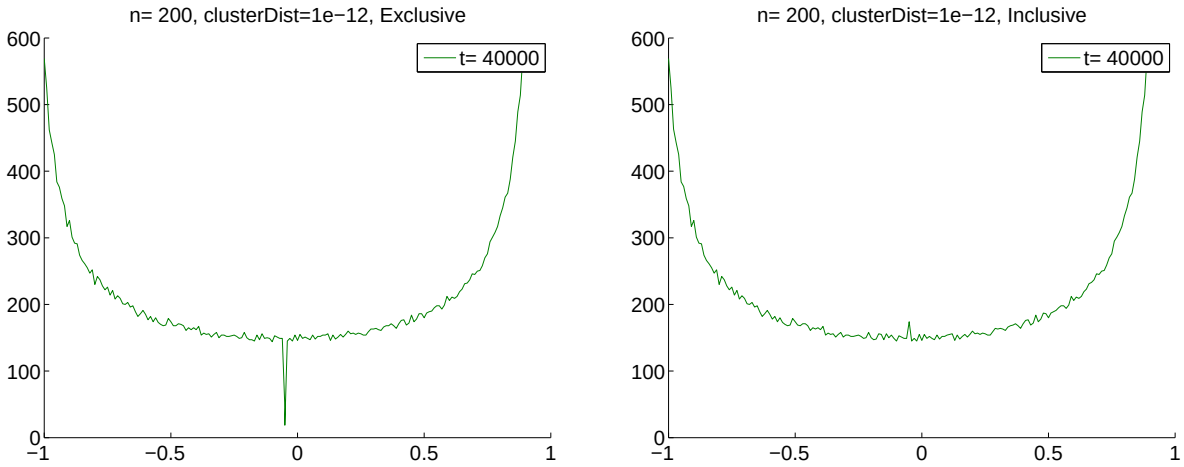

FIGURE 5.4. The number of Ritz values within 10^{-12} (left) and 10^{-8} (right) of each eigenvalue for a matrix of dimension $n = 200$ with a cluster of size 10 in the center of the spectrum spaced 10^{-12}. There are fewer Ritz values near each eigenvalue within the cluster than elsewhere, yet there are more Ritz values in the cluster area than if there was a single eigenvalue there.

$c \approx 1.5$, which is essentially the periodicity at which Ritz values appear near eigenvalues at the center of the spectrum. On small matrices ($n = 992$ in our experiment), $k = 2$ was almost always sufficient. For $n = 1737$, $k = 2$ was usually sufficient but sometimes convergence required $k = 3$ or $k = 4$. At $n = 5319$, convergence required up to $k = 5$ or $k = 6$.

8. Convergence on Real-World Matrices

Figures 8.1 and 8.2 explore the behavior of Lanczos on a large set of real-world matrices. We ran our code on a set of 133 matrices from Davis' University of Florida Sparse Matrix Collection This set includes all the symmetric matrices of dimension 2500 or less (and with numerical values; we omitted sparsity-pattern-only matrices). We did not attempt to disperse multiple eigenvalues; instead, we compared the eigenvalues computed by our code to those computed by LAPACK and counted how many agree to within $10^{-11}\|A\|$ or better; this tells when our code fails to find isolated eigenvalues or entire clusters, not when it converges to all the eigenvalues in a tight cluster.

The results show that Lanczos can compute all the eigenvalues of most of the matrices after $16n$ iterations; over 60% or them with 64-bit arithmetic, and more than 70% of them with 128- and 256-bit arithmetics. After only $4n$ or $8n$ iterations, Lanczos can still compute all the eigenvalues of many matrices. As we perform more iterations, Lanczos tends to resolve more eigenvalues that are *far away from already converged ones*. The remaining non-converged eigenvalues tend to get closer and closer to converged ones.

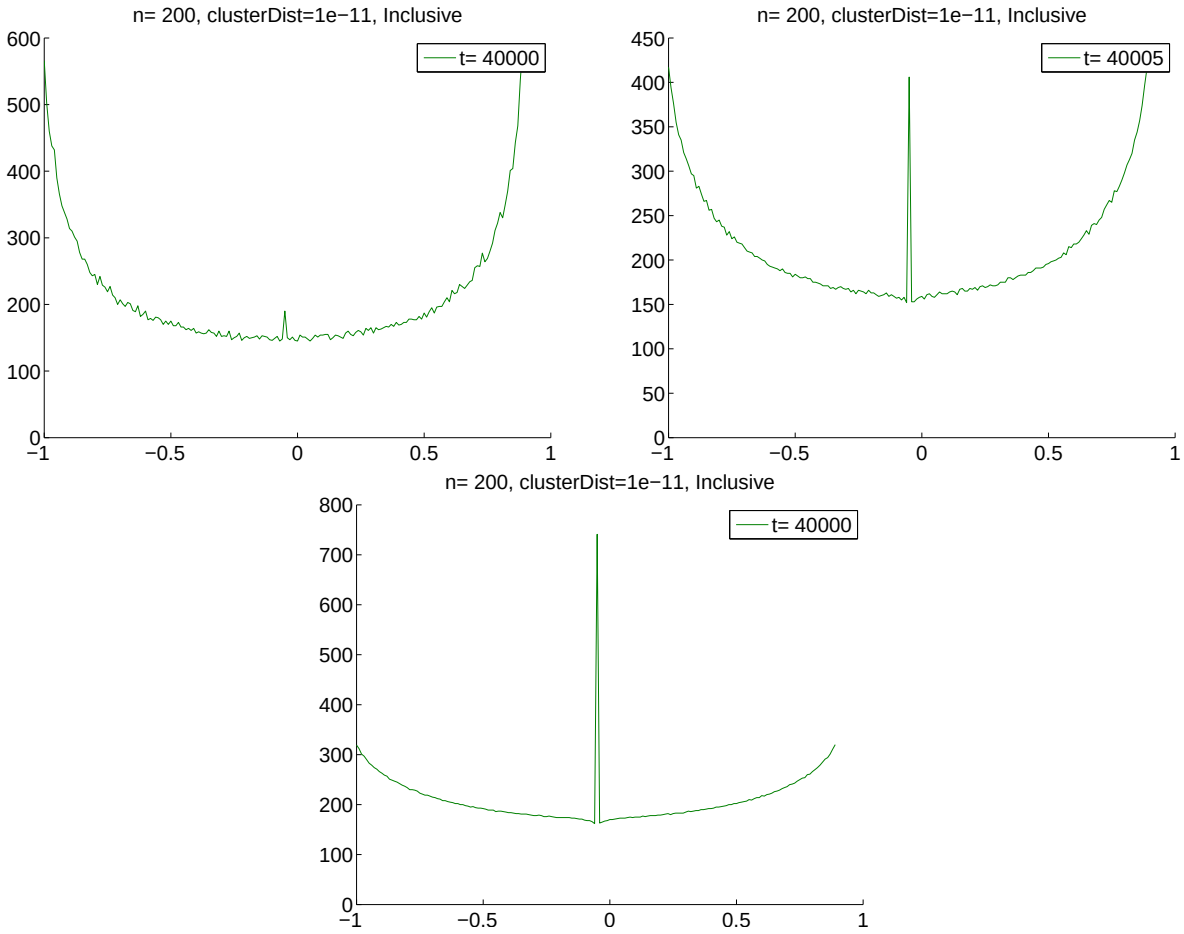

FIGURE 6.1. The number of Ritz values within 10^{-7} of each eigenvalue for a matrix of dimension $n = 200$ with a cluster of size 10 in the center of the spectrum spaced 10^{-11} using different floating-point precision: IEEE-754 double precision (64 bits; top left), 128 bit floating-point (top right) and 256 bit floating point (bottom).

9. Conclusions

Our experimental examination of the long-term behavior of the Lanczos process leads us to several conclusions.

First, as long as the distance between eigenvalues is large relative to $\epsilon_{\text{machine}}$, Lanczos converges to all the eigenvalues (and converges twice in roughly twice the number of iterations). If the minimal inter-eigenvalue distance is large relative to $\epsilon_{\text{machine}}$, the number of iterations m required for double convergence to all the eigenvalue grows very slowly as a multiple of n. In other words, m/n grows very slowly. On the other hand, as the distance between eigenvalues approaches $\epsilon_{\text{machine}}$, convergence slows down. These results are consistent with theoretical worst-case convergence rate bounds [9].

Large tight clusters of eigenvalues dramatically slow down the convergence of Lanczos to all the eigenvalues. Lanczos may be impractical for matrices with such spectra unless

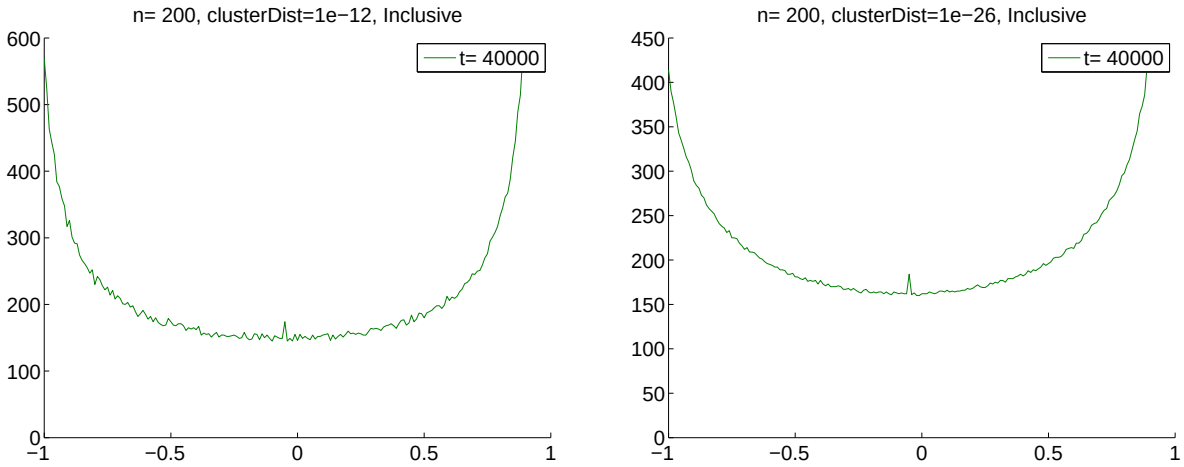

FIGURE 6.2. The number of Ritz values per unit length for a matrix of dimension $n = 200$ with a cluster of size 10 in the center of the spectrum spaced 10^{-12} using 64-bit floating-point arithmetic (left) for a matrix with a size 10 cluster spaced 10^{-26} with 128-bit floating point-precision arithmetic (right). In both cases the spacing between eigenvalues in the cluster is around $10^4 \epsilon_{\text{machine}}$.

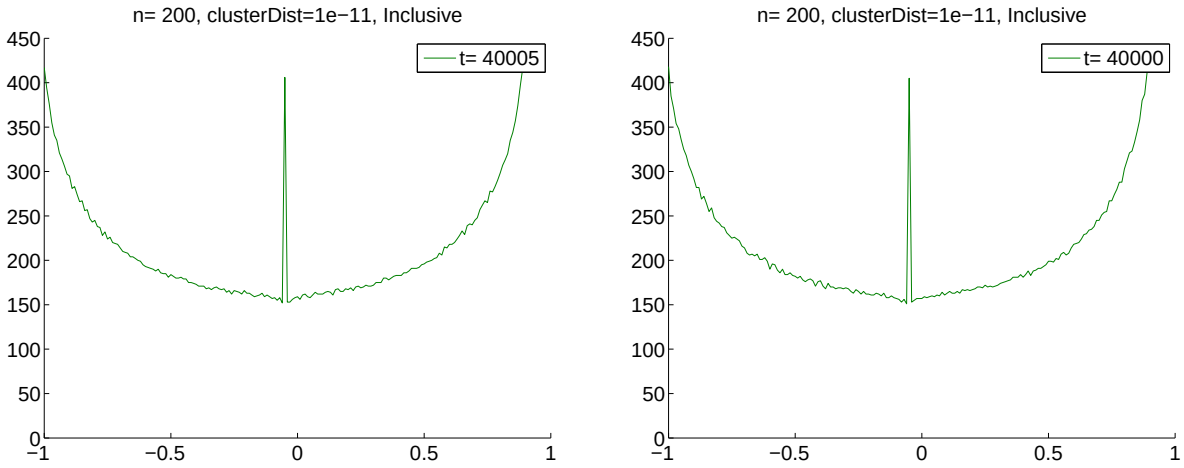

FIGURE 6.3. The number of Ritz values within 10^{-7} of each eigenvalue for a matrix of dimension $n = 200$ with a cluster of size 10 in the center of the spectrum, with eigenvalues spaced 10^{-11} apart. The graph on the left shows the behavior of Lanczos when the entire algorithm used 128-bit arithmetic, and the graph on the right shows the behavior when the tridiagonal matrix T was constructed using 128-bit arithmetic but its eigenvalues computed in 64-bit arithmetic.

measures are taken to address this issue. Initial experimentation on small matrices suggest that randomized dispersion is effective when the spectrum contains clusters but

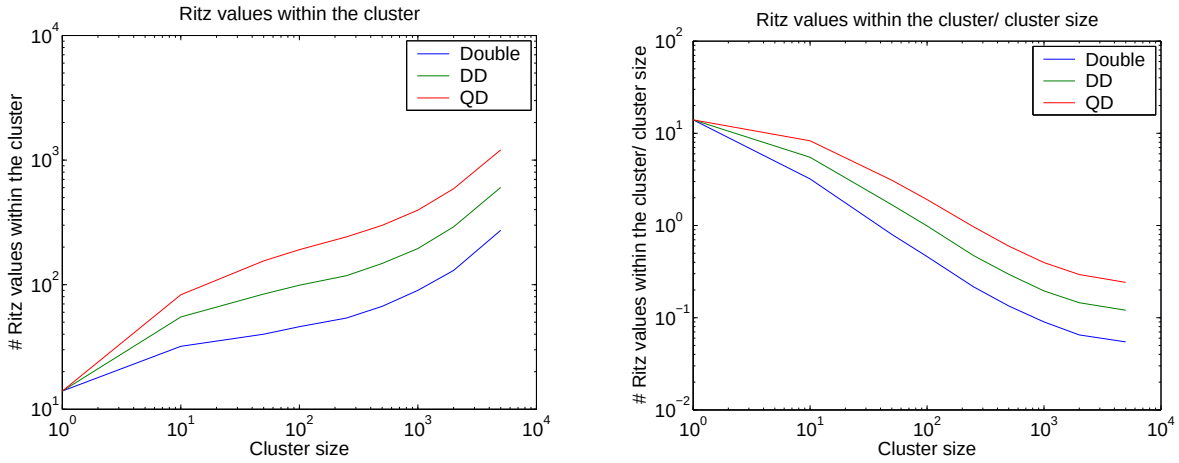

FIGURE 6.4. The number of Ritz values within a cluster of eigenvalues that are spaced 10^{-7} apart after $20n$ Lanczos iterations. Apart from the cluster, the spectrum contains 2000 eigenvalues spaced evenly between -1 and 1. In both graphs, the X axis shows the number of eigenvalues in the cluster, ranging from 1 to 5000. (The dimension of the matrices therefore ranged from 2000 to 7000). On the left, the Y axis shows the number of Ritz values in the cluster. On the right, the Y axis shows the same number, but divided by the size of the cluster. Both graphs show the results of computations in 64-bit arithmetic (double), 128-bit (DD), and 256-bit (QD).

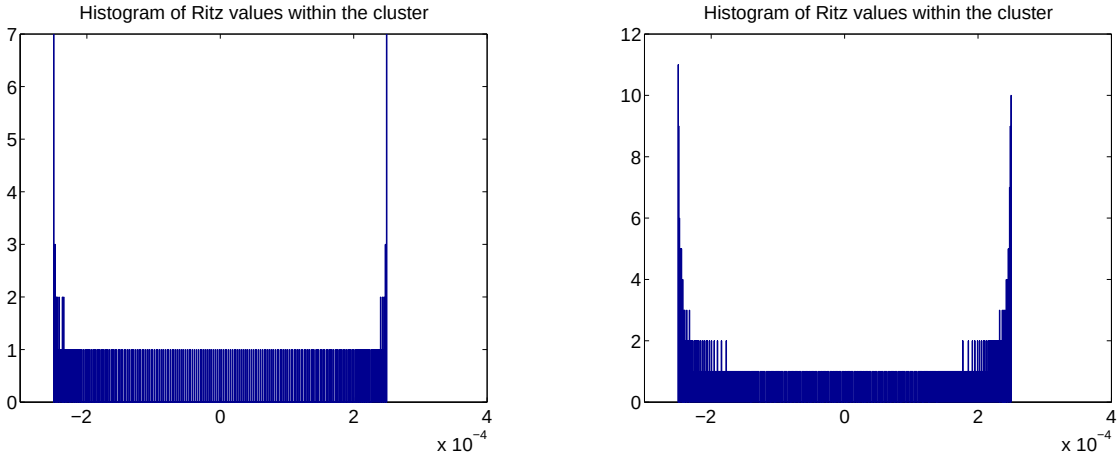

FIGURE 6.5. A histogram of the Ritz values within a cluster of 5000 eigenvalues that are spaced 10^{-7} apart after $20n = 20 \cdot 7000$ Lanczos iterations. The histogram on the left shows the results in 64-bit arithmetic and the results on the right in 128-bit.

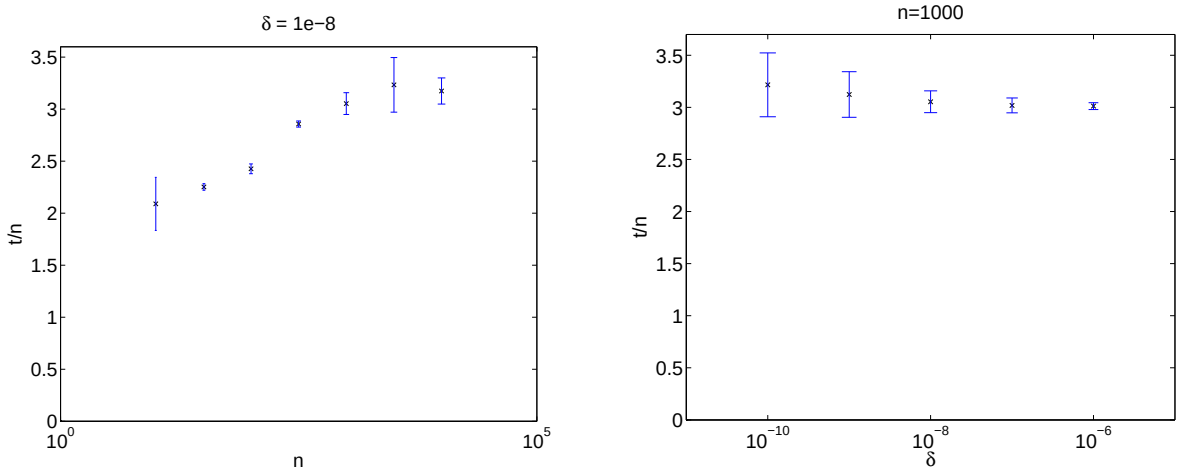

FIGURE 7.1. The number t of iterations required to achieve double convergence to all the eigenvalues for matrices with two clusters of eigenvalues at 1 and -1. On the left, the dimension n of A changes while the inter-eigenvalue distance δ is kept fixed; on the right, the dimension n is fixed and the distance δ varies. In both graphs, the Y axes are iteration counts t relative to the matrix dimension n. The black $\times$ symbols mark the average in 250 experiments with different starting vectors, and the blue lines span the standard deviations.

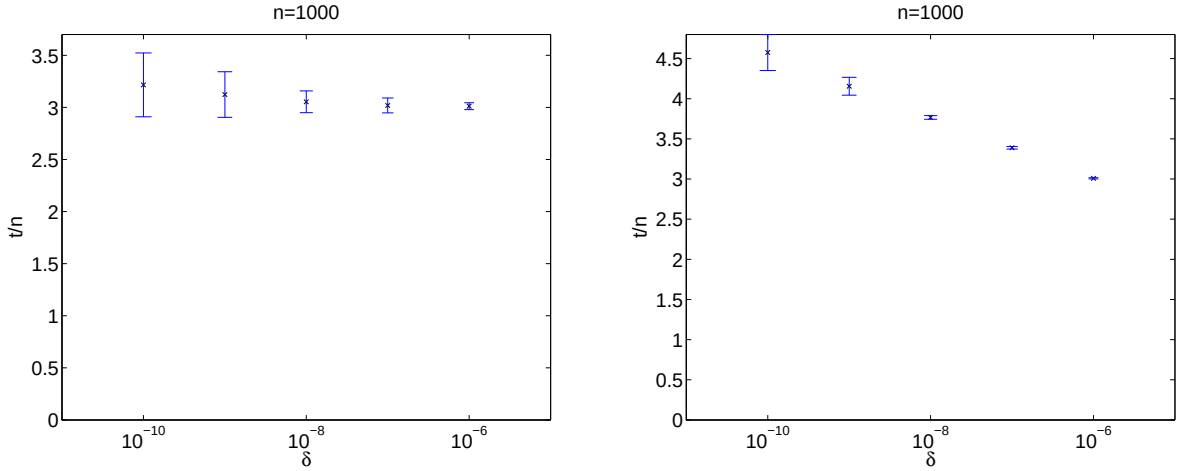

FIGURE 7.2. On the left, a graph from Figure 7.1. On the right, a similar experiment, but where the cluster at 1 had eigenvalues spaced 10^{-6} apart.

they are not too large, but ineffective when clusters are very large (say an eigenvalue of multiplicity 3000 in a matrix of dimension 10000).

We note that high-precision is not required to compute the Ritz values once T has been computed, as long as they are computed to a precision that allows convergence decisions to be made. If we are only interested in accelerating double convergence, it

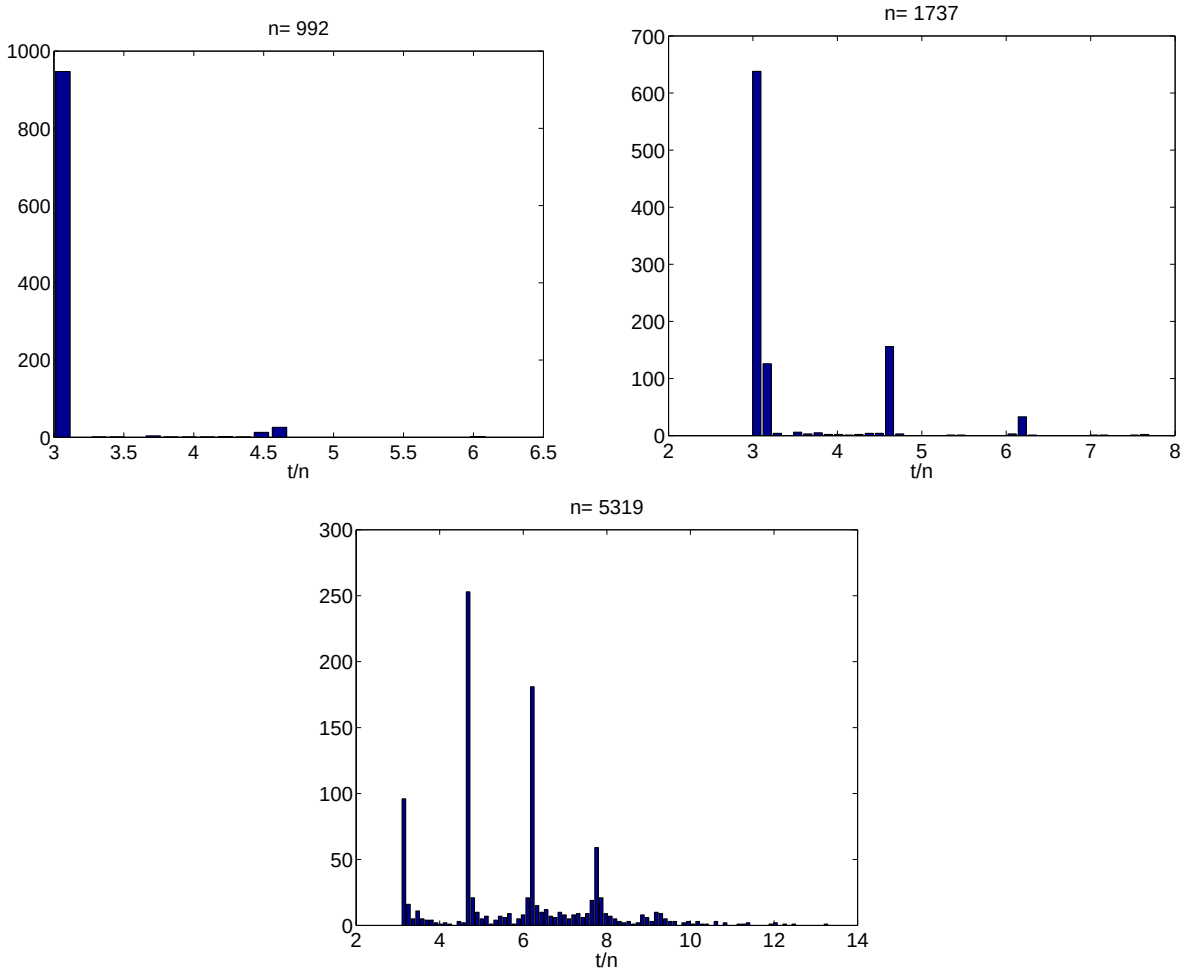

FIGURE 7.3. The number of runs out of 1000 in which there are at n Ritz clusters with Ritz values closer than 10^{-13} (indicating double convergence and hence detection of an eigenvalue). The matrices have regularly-spaced eigenvalues between -1 and 1. The runs differ only in the starting vector, which is random. As n grows, more iterations are required to obtain double convergence.

is enough to use high-precision arithmetic only in computations involving the Lanczos basis vectors. This observation is trivial given the perturbation theory of the eigenvalues of symmetric matrices and the stability of tridiagonal eigensolvers, but it is still useful in practice, since computing the Ritz values from $T^{(m)}$ (and computing $X^{(m)}$) is a computationally expensive part of Lanczos.

We have also found that Ritz values first converge but then slowly diverge. As far as we can tell, this observation is completely new. Therefore, double convergence (perhaps any valid convergence criterion) becomes less useful if we apply it after too many iterations.

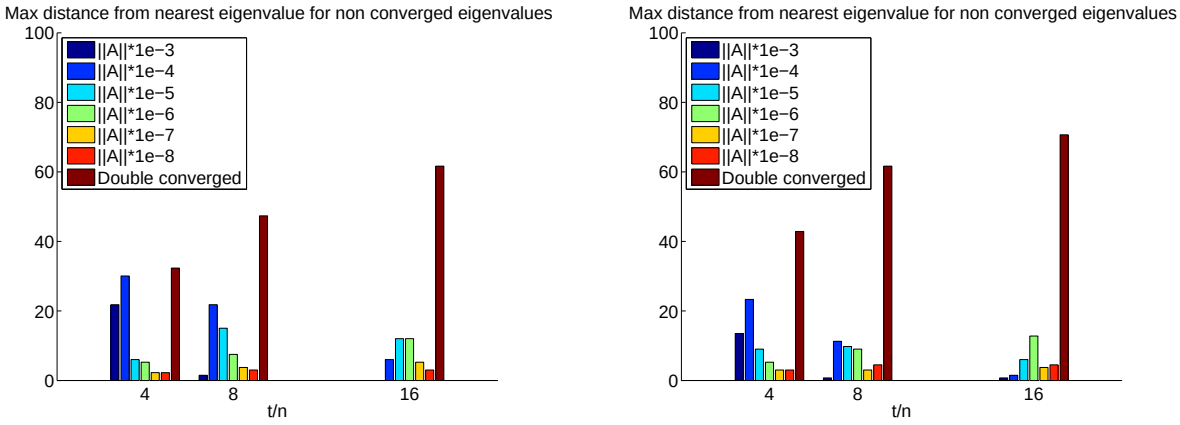

FIGURE 8.1. Convergence behavior on a set of 133 real-world matrices. The graphs show the percentage of matrices that have converged to all the eigenvalues after $4n$, $8n$, and $16n$ iterations (in brown). The code did not attempt to find the multiplicity of each eigenvalue. The graphs also show, for matrices that have not converged, the minimum distance from a converged eigenvalue to a non-converged one. The graph on the right shows results for 64-bit computations, and the graph on the left for 128-bit computations. In high precision, T was computed in high precision but its eigenvalues were computed in 64-bit arithmetic.

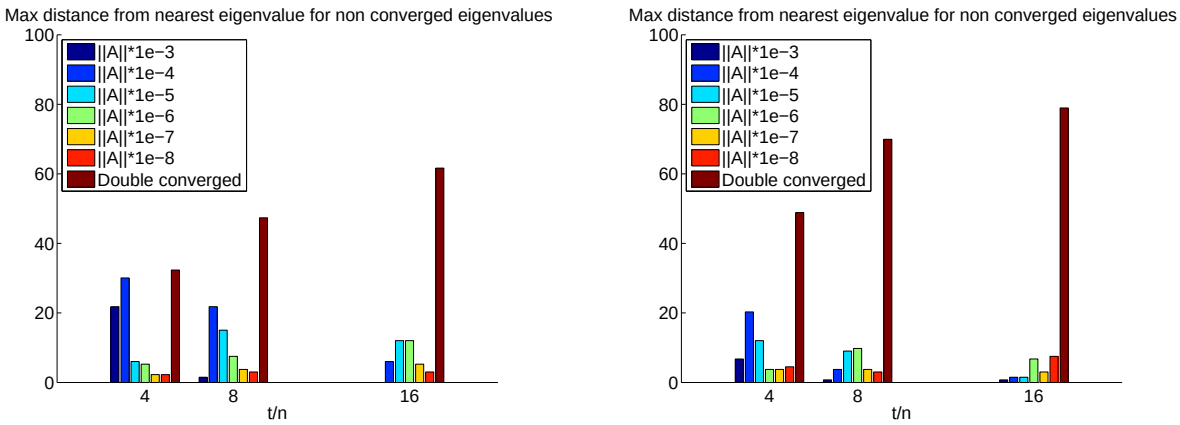

FIGURE 8.2. Convergence behavior on real-world matrices, using 64-bit arithmetic (left) and 256-bit arithmetic (right). The setup is the same as in Figure 8.1.

Finally, we have found that Lanczos can find *all* the eigenvalues of many large real-world matrices.

REFERENCES

[1] D. Calvetti, L. Reichel, and D.C. Sorensen. An implicitly restarted Lanczos method for large symmetric eigenvalue problems. *Electronic Transactions on Numerical Analysis*, 2:1–21, 1994.

[2] Jane K. Cullum and Ralph A. Willoughby. *Lanczos Algorithms for Large Symmetric Eigenvalue Computations: Vol. 1 Theory*. Birkäuser, 1985.
[3] V. L. Druskin and L. A. Knizhnerman. Error bounds in the simple Lanczos procedure for computing functions of symmetric matrices and eigenvalues. *USSR Computational Mathematics and Mathematical Physics*, 31(7):20–30, 1991.
[4] J. T. Edwards, D. C. Licciardello, and Thouless D. J. Use of the Lanczos method for finding complete sets of eigenvalues of large sparse symmetric matrices. *Journal of the Institute of Mathematics and its Applications*, 23:277–283, 1979.
[5] Gene H. Golub and Charles F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, 3rd edition, 1996.
[6] Joseph F. Grcar. *Analyses of the Lanczos Algorithm and of the Approximatino Problem in Richardson's Method*. PhD thesis, University of Illinois at Urbana-Champaign, December 1981.
[7] A. Greenbaum. Behavior of slightly perturbed Lanczos and conjugate-gradient recurrences. *Linear Algebra and its Applications*, 113:7–63, 1989.
[8] Thomas Kalkreuter. Study of Cullum's and Willoughby's Lanczos method for Wilson fermions. *Computer Physics Communications*, 95:1–16, 1996.
[9] L. A. Knizhnerman. The quality of approximations to a well-isolated eigenvalue, and the arrangement of "Ritz numbers" in a simple Lanczos process. *Computational Mathematics and Mathematical Physics*, 35(10):1175–1187, 1995.
[10] Cornelius Lanczos. An iteration method for the solution of the eigenvalue problem of linear differential and integral operators. *Journal of Research of the National Bureau of Standards*, 45(4):255–282, October 1950.
[11] Gérard Meurant. *The Lanczos and Conjugate Gradient Algorithms: From Theory to Finite Precision Computations*. SIAM, 2006.
[12] C. C. Paige. Accuracy and effectiveness of the Lanczos algorithm for the symmetric eigenproblem. *Linear Algebra and its Applications*, 34:235–258, 1980.
[13] B. N. Parlett and J. K. Reid. Tracking the progress of the Lanczos algorithm for large symmetric eigenproblems. *IMA Journal of Numerical Analysis*, 1:135–155, 1981.
[14] B. N. Parlett and D. S. Scott. The Lanczos algorithm with selective orthogonalization. *Mathematics of Computation*, 33:217–238, 1979.
[15] Beresford Parlett. Misconvergence in the Lanczos algorithm. In M.G. Cox and S. Hammarling, editors, *Reliable Numerical Computation*, pages 7–24. Clarendon Press, Oxford, 1990.
[16] Beresford N. Parlett. *The Symmetric Eigenvalue Problem*. Prentice-Hall, 1980.
[17] Yousef Saad. *Numerical Methods for Large Eigenvalue Problems*. 2nd edition, 2011.
[18] Horst D. Simon. Analysis of the symmetric Lanczos algorithm with reorthogonalization methods. *Linear Algebra and its Applications*, 61:101–131, 1984.
[19] Horst D. Simon. The Lanczos algorithm with partial reorthogonalization. *Mathematics of Computation*, 42:115–142, 1984.
[20] G. W. Stewart. *Matrix Algorithms, Volume 2: Eigensystems*. SIAM, 2001.
[21] Kesheng Wu and Horst Simon. Thick-restart Lanczos method for large symmetric eigenvalue problems. *SIAM Journal on Matrix Analysis and Applications*, 22:602–616, 2000.
[22] Wolfgang Wülling. On stabilization and convergence of clustered Ritz values in the Lanczos method. *SIAM Journal on Matrix Analysis and Applications*, 27:891–908, 2006.