Eigenvalue Computations: The Power and Lanczos Methods

by Mark MacLean
North Carolina State University

Introduction

This paper looks at two methods for finding the eigenvalues of a matrix. The first section examines the behavior of symmetric and nonsymmetric matrices under the power method. Differences in the convergence rates of these two classes of matrices are presented and explained. The second section focuses on the extreme eigenvalues computed by the Lanczos algorithm. In particular, the paper examines error bounds and approximations for the symmetric and unsymmetric Lanczos methods.

I. The Power Method

The power method is known as a simple iterative process for finding the dominant eigenvalue of a matrix. The method proceeds as follows: Let x_0 be an arbitrary initial vector, and let $A \in \mathbb{R}^{n \times n}$ have eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_n$ such that $|\lambda_1| \leq |\lambda_2| \leq \cdots < |\lambda_n|$. Construct the sequence $x_{k+1} = Ax_k$ for $k = 0, 1, 2, \ldots.$ Assuming the dominant eigenvalue λ_n exists, this sequence converges to a multiple of the associated eigenvector. The method is easily explained by expanding x_0 in terms of the eigenvectors of A. Suppose the eigenvectors $v_1, v_2, \ldots, v_n$ of the matrix are linearly independent; that is, A is non-defective.[1] Then x_0 can be written as

$$
\begin{aligned}
x_0 &= \alpha_1 v_1 + \cdots + \alpha_n v_n \\
\text{Thus,} \quad x_k &= A^k x_0 = \alpha_1 \lambda_1^k v_1 + \cdots + \alpha_n \lambda_n^k v_n \\
&= \lambda_n^k [\alpha_1 (\frac{\lambda_1}{\lambda_n})^k v_1 + \cdots + \alpha_{n-1} (\frac{\lambda_{n-1}}{\lambda_n})^k v_{n-1} + \alpha_n v_n] \\
&\to \alpha_n \lambda_n^k v_n \quad \text{as } k \to \infty
\end{aligned}
\tag{1}
$$

When using the power method, the x_k's are usually scaled at each step to prevent underflow or overflow from the λ_n^k term. Also, an approximation to the dominant eigenvalue can be computed at each step from the Rayleigh quotient:

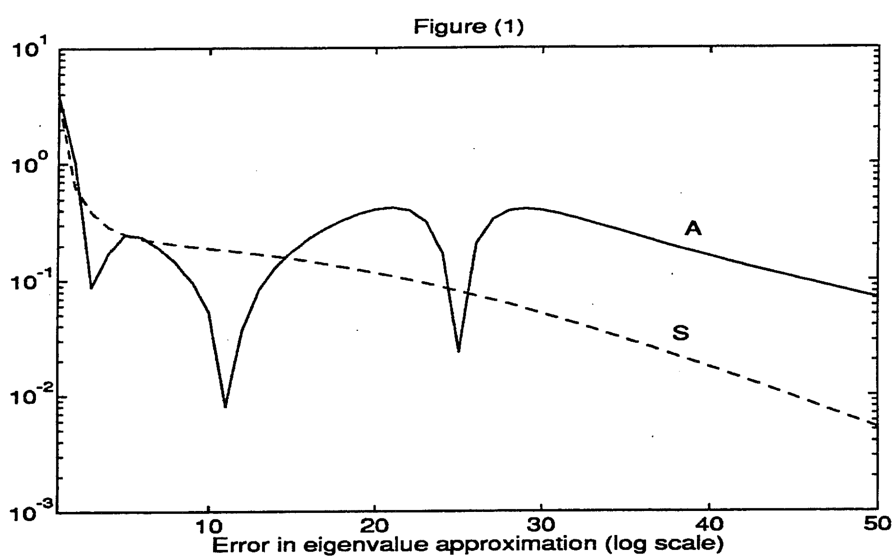
$$\mu_k = \frac{x_k^T A x_k}{x_k^T x_k} \tag{2}$$

[1]Even if A is defective, the power method can still yield the dominant eigenvector if it has multiplicity one.

As x_k approaches the dominant eigenvector, the Rayleigh quotient μ_k will be an increasingly better estimate of the dominant eigenvalue. Equation (1) suggests that the speed of convergence of the power method is determined by the ratio $\frac{\lambda_{n-1}}{\lambda_n}$. Many of the linear algebra textbooks discuss this point no further. However, there is more to the convergence of the power method than the ratio $\frac{\lambda_{n-1}}{\lambda_n}$. Figure (1) plots the power method's error for two matrices, A and S, that have exactly the same set of eigenvalues. Regarding this graph, one can see that a couple of unexpected things occur. Not only does S converge much faster than A, but the eigenvalue estimations of A appear to jump up and down initially whereas the convergence of S is very smooth. The differences in convergence occur because S is a symmetric matrix. The comparison of the behavior of symmetric versus nonsymmetric matrices under the power method will be the focus of this section.

The tridiagonal matrices with the same eigenvalues from Figure (1) are found below. These matrices have exactly the same set of eigenvalues since S is just a similarity transformation of A, i.e. $A = D^{-1}SD$, where D is the diagonal matrix with diagonal components $(2^0, 2^1, ..., 2^9)$.

$$A_{10\times 10} = \begin{bmatrix} 2 & -2 & & 0 \\ \frac{-1}{2} & 2 & \ddots & \\ & \ddots & \ddots & -2 \\ 0 & & \frac{-1}{2} & 2 \end{bmatrix} \qquad S_{10\times 10} = \begin{bmatrix} 2 & -1 & & 0 \\ -1 & 2 & \ddots & \\ & \ddots & \ddots & -1 \\ 0 & & -1 & 2 \end{bmatrix}$$



This figure plots the error in the power method's eigenvalue approximation for 50 iterations applied to A and S.

A good heuristic explanation of A's initial oscillation can be given in terms of an orthogonal expansion of the x_k's. Since S is symmetric, an orthogonal set of eigenvectors exists.

We can write x_0 as a linear combination of these eigenvectors:

$$x_0 = \alpha_1 v_1 + \cdots + \alpha_{10} v_{10} = V\alpha, \qquad \alpha = \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_{10} \end{bmatrix}$$

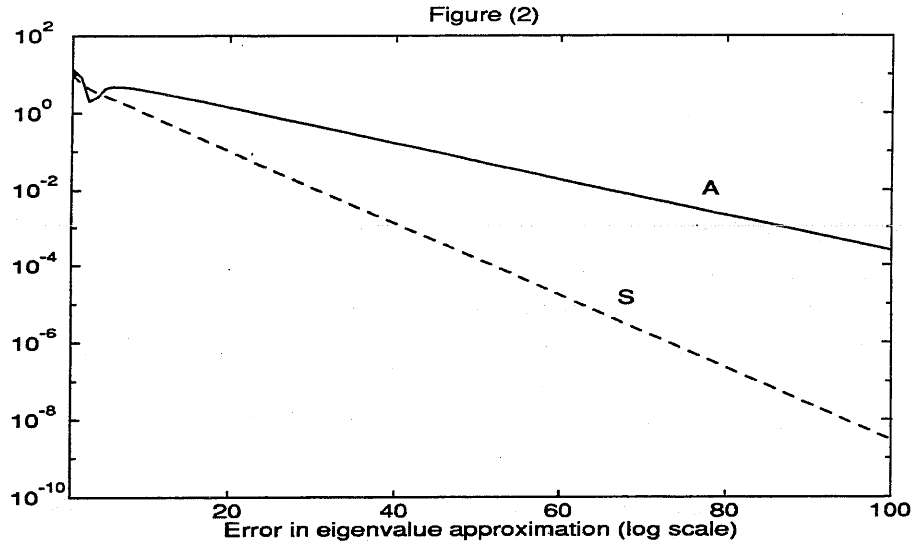
Since V is an orthogonal matrix, it is length preserving, and $\|x_0\| = \|\alpha\|$. The coefficients α_i are on the same scale as the components of x_0. On the other hand, the unsymmetric matrix A has eigenvectors that are not orthogonal. Let $x_0 = \beta_1\omega_1 + \cdots + \beta_{10}\omega_{10}$, where ω_i is an eigenvector of A. These coefficients β_i can be rather large, making $\|\beta\| \gg \|x_0\|$. For instance, let $x_0 \in \mathbb{R}^{10}$ be the vector $\frac{1}{\sqrt{10}}(1, \ldots, 1)^T$. Then by writing x_0 as an expansion of each set of eigenvectors, we find that $\|\alpha\| = \|x_0\| = 1$, but $\|\beta\| \approx 69.5$. When β has such large positive or negative components for successive x_k's, we see the Rayleigh quotient oscillate as in Figure (1). Only when x_k is close enough to the dominant eigenvector does the norm of β approximate the norm of x_k and the convergence curve becomes smooth. This is easiest to see if we normalize all the eigenvectors and x_k's to be of unit length. Then as x_k approaches ω_{10}, β_{10} approaches 1 but all the other coefficients go to zero. Thus, $\|\beta\| \approx \|x_k\| = 1$.

Figure (2) illustrates the difference in convergence rates for another pair of 10×10 matrices sharing the same eigenvalues:

$$A_1 = \begin{bmatrix} -5 & 2 & & & & 0 \\ \frac{1}{2} & -7 & 3 & & & \\ & \frac{1}{3} & -9 & 4 & & \\ & & \frac{1}{4} & -11 & \ddots & \\ & & & \ddots & \ddots & 10 \\ 0 & & & & \frac{1}{10} & -23 \end{bmatrix} \qquad S_1 = \begin{bmatrix} -5 & 1 & & & & 0 \\ 1 & -7 & 1 & & & \\ & 1 & -9 & 1 & & \\ & & 1 & -11 & \ddots & \\ & & & \ddots & \ddots & 1 \\ 0 & & & & 1 & -23 \end{bmatrix}$$

The difference can be explained by properties of the Rayleigh quotient (equation (2)) applied to symmetric matrices. Calculating Rayleigh quotients for a symmetric matrix has the effect of squaring the error in the initial vector x_k. That is, suppose $x_k = v_n + e$, where $\| e \| = \epsilon$ and v_n is an eigenvector of A. Then $\mu_k = \lambda_n + O(\epsilon^2)$. A proof may be found in Stewart [5]. Thus, while the eigenvalues for A converge linearly at the same rate as the eigenvectors of both matrices, the eigenvalues of S converge twice as fast. While this is still extremely slow, the difference is worth noting.

The properties discussed are not unique to the two matrix pairs A and S given here. Other examples of symmetric and nonsymmetric matrices sharing the same eigenvalues can easily be constructed on mathematical software. For example, on Matlab, let $\mathbf{A} = \mathbf{rand(n)}$, and find the eigensystem of A by $\mathbf{[V,E] = eig(A)}$, where the columns of V are eigenvectors and the diagonal elements of E are the eigenvalues. The command $\mathbf{B = orth(V)}$ gives an orthogonal basis for the eigenvectors, and $\mathbf{S = inv(B) * E * B}$ generates a symmetric matrix with the same eigenvalues of A. Initial vectors x_0 should be randomly chosen. Graphs of the convergence of the power method for these matrices will parrot the properties described.



This figure plots the error in the eigenvalue approximation for 100 iterations of the power method applied to matrices A and S.

II. Lanczos

The Lanczos method for finding eigenvalues of a symmetric matrix is typically applied to large, sparse matrices when a few of the extreme eigenvalues are desired. Given a symmetric matrix $A \in \mathbb{R}^{n\times n}$, the method computes a sequence of tridiagonal matrices T_j whose largest and smallest eigenvalues are usually close approximations of the extreme eigenvalues of A. The transformations are made with orthonormal, $n \times j$ matrices Q_j such that $Q_j^T A Q_j = T_j$ ($T_n \equiv T$). Thus, when $j = n$, A and T have exactly the same set of eigenvalues since this is just a similarity transformation. The transformation is useful because the eigenspace of T is much easier to compute than that of A. This method was developed by Cornelius Lanczos in 1950 as a complete algorithm for tridiagonalization. In actual use, round-off errors cause the method to break down as the orthogonality of the columns of Q is lost. However, the practicality of the Lanczos algorithm is that the extreme eigenvalues of A are closely approximated when j is much less than n. This section will focus on the error bounds and approximations of the eigenvalues of the T_j's.

<u>The Lanczos method</u> (Golub and Van Loan [2])
Let $A \in \mathbb{R}^{n\times n}$ be symmetric. We want to find an orthogonal matrix Q such that

$$Q^T A Q = T = \begin{bmatrix} \alpha_1 & \beta_1 & & & 0 \\ \beta_1 & \alpha_2 & \beta_2 & & \\ & \beta_2 & \alpha_3 & \ddots & \\ & & \ddots & \ddots & \beta_{n-1} \\ 0 & & & \beta_{n-1} & \alpha_n \end{bmatrix} \tag{3}$$

To show how this can be done, we first represent Q by its columns: $Q = [q_1\, q_2\, \ldots\, q_n]$. Since $AQ = QT$, we can set the j^{th} column of AQ equal to the j^{th} column of QT. Thus,

$$Aq_j = \beta_{j-1}q_{j-1} + \alpha_j q_j + \beta_j q_{j+1} \;\; (\beta_0 q_0 \equiv 0) \;\; j = 1 : n-1 \tag{4}$$

Since the columns of Q are mutually orthogonal, multiplying equation (4) by q_j^T yields $\alpha_j = q_j^T A q_j$. Let $r_j = (A - \alpha_j I)q_j - \beta_{j-1}q_{j-1}$. Then $q_{j+1} = r_j/\beta_j$, where $\beta_j = \pm \; \|r_j\|_2$. If $r_j = 0$ the algorithm terminates prematurely. We have thus written explicit formulas for the α_j and β_j. This method may be programmed into Matlab as follows:

```
Matrix = input('Matrix = ?');
q(:,1) = zeros(size(Matrix,1),1);
q(:,2) = input('initial vector = ?');
j = 0;
u(1) = 1;
r(:,1) = q(:,2);
np = size(Matrix,1);
for j = 2:(np+1)
d(j) = q(:,j)'*Matrix*q(:,j);
r(:,j) = (Matrix - d(j)*eye(np))*q(:,j) - u(j-1)*q(:,j-1);
u(j) = norm(r(:,j));
q(:,j+1) = r(:,j)/u(j);
end;
Tridiag = zeros(np,np);
for j = 1:size(Matrix,1)-1
Tridiag(j,j) = d(j+1);
Tridiag(j+1,j) = u(j+1);
Tridiag(j,j+1) = u(j+1);
end
Tridiag(j+1,j+1) = d(j+2);
```

A Lanczos algorithm for unsymmetric matrices exists but is not very practical. Given $A \in \mathbb{R}^{n \times n}$, one can find an invertible (but not necessarily orthogonal) matrix X such that

$$X^{-1}AX = T = \begin{bmatrix} \alpha_1 & \gamma_1 & & & 0 \\ \beta_1 & \alpha_2 & \gamma_2 & & \\ & \beta_2 & \alpha_3 & \ddots & \\ & & \ddots & \ddots & \gamma_{n-1} \\ 0 & & & \beta_{n-1} & \alpha_n \end{bmatrix} \tag{5}$$

Just like the symmetric case, the extreme eigenvlaues of A are located fairly quickly by a small submatrix of T. As pointed out in Golub and Van Loan [2], the tridiagonalization of a nonsymmetric matrix is a very unstable process. The unsymmetric Lanczos method is almost never used in practice,although some recent research has attempted to improve this algorithm, such as Parlett, Taylor and Liu's "look-ahead Lanczos algorithm" [3]. The Matlab program for this algorithm is as follows:

```
Matrix = input('Matrix = ?');
x(:,2) = input('initial vector x = ?');
y(:,2) = x(:,2);
x(:,2) = x(:,2)/norm(x(:,2));
y(:,2) = y(:,2)/norm(y(:,2));
N = size(Matrix,1);
x(:,1)=zeros(N,1); y(:,1)=x(:,1);
c(1) = 0; b(1) = 0;
for j = 2:N
a(j) = y(:,j)'*Matrix*x(:,j);
r(:,j) = (Matrix - a(j)*eye(N))*x(:,j) - c(j-1)*x(:,j-1);
b(j) = norm(r(:,j));
x(:,j+1) = r(:,j)/b(j);
p(:,j) = (Matrix - a(j)*eye(N))'*y(:,j) - b(j-1)*y(:,j-1);
c(j) = x(:,j+1)'*p(:,j);
y(:,j+1) = p(:,j)/c(j);
end
a(N+1) = y(:,N+1)'*Matrix*x(:,N+1);
Tridiag = zeros(N,N);
for j = 1:N-1
Tridiag(j,j) = a(j+1);
Tridiag(j+1,j) = b(j+1);
Tridiag(j,j+1) = c(j+1);
end
Tridiag(N,N) = a(N+1);
```

Note: This program does not work for complex matrices.

It has been mentioned that the sequence T_j tends to locate the extreme eigenvalues of A very quickly. The Kaniel-Page theorem gives a lower bound for the largest and smallest eigenvalues of T_j in terms of the eigenvalues of A.

Kaniel-Paige Theorem. (Golub and Van Loan [2])
Let A be an $n \times n$ symmetric matrix with eigenvalues $\lambda_1 \leq \ldots \leq \lambda_n$ and corresponding orthonormal eigenvectors $v_1, \ldots, v_n$. Let $\theta_1 \leq \ldots \leq \theta_j$ be the eigenvalues of the matrix T_j obtained after j steps of the Lanczos method. Then

$$\lambda_n \geq \theta_j \geq \lambda_n - \frac{(\lambda_n - \lambda_1)\tan(\phi_n)^2}{(c_{j-1}(1+2\rho_n))^2} \tag{6}$$

where $\cos(\phi_n) = | q_1^T v_n |$, $\rho_n = (\lambda_n - \lambda_{n-1})/(\lambda_{n-1} - \lambda_1)$, *and* $c_{j-1}(x)$ *is the Chebyshev polynomial of degree* $j-1$. *Similarly,*

$$\lambda_1 \le \theta_1 \le \lambda_1 + \frac{(\lambda_n - \lambda_1)\tan(\phi_1)^2}{(c_{j-1}(1+2\rho_1))^2} \tag{7}$$

where $\cos(\phi_1) = | q_1^T v_1 |$ and $\rho_1 = (\lambda_2 - \lambda_1)/(\lambda_n - \lambda_2)$.

Proof. (Golub and Van Loan [2])
From the minimax theorem of eigenvalues, we have

$$\theta_j = \max_{y \ne 0} \frac{y^T T_j y}{y^T y} = \max_{y \ne 0} \frac{y^T (Q_j^T A Q_j) y}{y^T y} = \max_{y \ne 0} \frac{(Q_j y)^T A (Q_j y)}{(Q_j y)^T (Q_j y)}$$

Let $w = Q_j y$. Then $w \in \operatorname{span}\{q_1, q_2, \ldots, q_j\}$. One can show by induction that $w \in \operatorname{span}\{q_1, Aq_1, \ldots, A^{j-1}q_1\}$, providing that the Lanczos process does not break down. This implies that $w = p(A)q_1$ for $p \in P^{j-1}$, where P^{j-1} is the set of polynomials of degree $j-1$. Then $w = Vp(\Lambda)V^T q_1$ since $A = V\Lambda V^T$ by the spectral theorem.

Thus, $\theta_j = \max\limits_{w \in K(A, q_1, j)} \dfrac{w^T A w}{w^T w}$ where K is the Krylov subspace $\equiv \operatorname{span}\{q_1, Aq_1, \ldots, A^{j-1}q_1\}$.

Since λ_n is the maximum of $\frac{w^T A w}{w^T w}$ over all nonzero w, it follows that $\lambda_n \ge \theta_j$. Now let $d = V^T q_1$ where d is just the vector of coefficients if we expand q_1 as a linear combination of A's eigenvectors. Then

$$\begin{aligned}
\theta_j &= \max_{p \in P^{j-1}} \frac{q_1^T V p(\Lambda)^T \Lambda p(\Lambda) V^T q_1}{q_1^T V p(\Lambda)^T p(\Lambda) V^T q_1} = \max_{p \in P^{j-1}} \frac{d^T p(\Lambda)^T \Lambda p(\Lambda) d}{d^T p(\Lambda)^2 d} \\
&= \max_{p \in P^{j-1}} \frac{\sum_{i=1}^n d_i^2 p(\lambda_i)^2 \lambda_i}{\sum_{i=1}^n d_i^2 p(\lambda_i)^2} && (8) \\
&\ge \max_{p \in P^{j-1}} \lambda_n - \frac{(\lambda_n - \lambda_1)\sum_{i=1}^{n-1} d_i^2 p(\lambda_i)^2}{d_n^2 p(\lambda_n)^2 + \sum_{i=1}^{n-1} d_i^2 p(\lambda_i)^2} && (9)
\end{aligned}$$

Note that this bound holds for any *specific* polynomial $p(x)$ we choose. We can make a tight bound by choosing our polynomial $p(x)$ that is large at λ_n and relatively small for the non-dominant eigenvalues. Let

$$p(x) = c_{j-1}(-1 + 2\frac{x - \lambda_1}{\lambda_{n-1} - \lambda_1})$$

where c_{j-1} is the Chebyshev polynomial defined by $c_j(x) = \cos(j cos^{-1} x)$. For $x \in [-1, 1]$, $| c_j(x) | \le 1$; but the Chebyshev polynomial grows very rapidly outside this interval. Thus, $| p(\lambda_i) | \le 1$ for $i = 1 : n-1$, while

$$p(\lambda_n) = c_{j-1}(1 + 2\rho_n), \rho_n = \frac{\lambda_n - \lambda_{n-1}}{\lambda_{n-1} - \lambda_1}.$$

Since $\|d\|= 1$,

$$1 - d_n^2 = \sum_{i=1}^{n-1} d_i^2 \geq \sum_{i=1}^{n-1} d_i^2 p(\lambda_i)^2.$$

$$\text{Thus,}\ \ \theta_j \geq \lambda_n - \frac{(\lambda_n - \lambda_1)(1 - d_n^2)}{d_n^2 c_{j-1}(1 + 2\rho_n)^2}$$

To obtain the final bound of the Kaniel-Paige theorem, note that $\tan(\phi_n)^2 = (1 - d_n^2)/d_n^2$.□

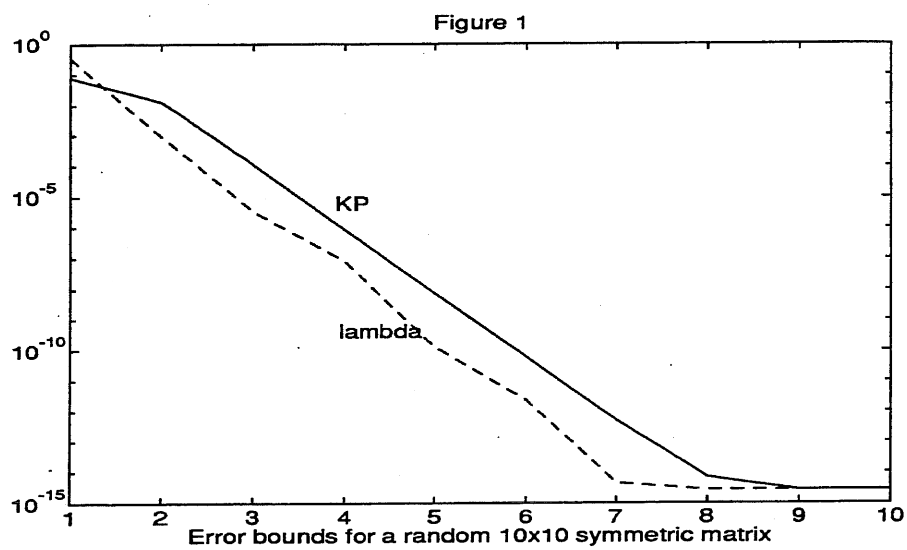


Figure (1) graphs the proximity of the theorem's upper and lower bounds to the dominant eigenvalue for successive T_j's. One can see that for random matrices, the upper bound, λ_n, is actually a closer approximation of θ_j than the lower bound (denoted KP).

The proof suggests an interesting result that occurs when there is a multiplicity of the eigenvalues. If A has k distinct eigenvalues where $k < n$, then the extreme eigenvalues θ_1 and θ_k of the matrix T_k generated by Lanczos will be exactly equal to λ_1 and λ_n. To see why this is true, look at equation (8). The polynomial $p(x)$ in this case has degree $k - 1$. Since

$$\lambda_n \geq \theta_k \geq \frac{\sum_{i=1}^n d_i^2 p(\lambda_i)^2 \lambda_i}{\sum_{i=1}^n d_i^2 p(\lambda_i)^2} \text{ for any } p \in P^{k-1},$$

we can choose our polynomial $p(x)$ to have roots at $\lambda_1, \ldots, \lambda_{k-1}$. Then $\lambda_n \geq \theta_k \geq \lambda_n$, so $\theta_k = \lambda_n$. One can show $\theta_1 = \lambda_1$ in the same manner.

Very little has been written about the unsymmetric Lanczos method. The author is unaware of any error bounds or approximations for the eigenvalues of T comparable to the Kaniel-Paige theorem for the symmetric case. In 1987, Cybenko [1] published an explicit formula for the characteristic polynomials of the T_j's in terms of the eigenspace of A. This formula applies to all nondefective matrices A.

Cybenko's Theorem. *Let λ_i, u_i, v_i be the eigenvalues, right eigenvectors, and left eigenvectors of A respectively. Let x and y be arbitrary vectors, and assume A is nondefective. The characteristic polynomial $c_j(\lambda)$ of the j^{th} approximating matrix obtained in the Lanczos process satisfies*

$$c_j(\lambda) = \sigma \sum_{I \in \Delta nj} \gamma_{i_1}\gamma_{i_2}\cdots\gamma_{i_j} V^2(\lambda_{i_1}, \lambda_{i_2}, \ldots, \lambda_{i_j}) \times (\lambda - \lambda_{i_1})(\lambda - \lambda_{i_2})\cdots(\lambda - \lambda_{i_j})$$

where σ is a constant and V is the Vandermonde determinant. The factors γ_j are

$$\gamma_j = (y^* u_j)(v_j^* x)$$

See Cybenko [1] for the proof.

Cybenko's theorem sheds more light on the case of the multiple eigenvalues. If A has k distinct eigenvalues, then not only are the extreme eigenvalues of T_k exact, but all of the eigenvalues are exact.

Proof. The characteristic polynomial of T_k can be written as a sum of terms of the form

$$\gamma_{i_1}\gamma_{i_2}\cdots\gamma_{i_j} V^2(\lambda_{i_1}, \lambda_{i_2}, \ldots, \lambda_{i_j}) \times (\lambda - \lambda_{i_1})(\lambda - \lambda_{i_2})\cdots(\lambda - \lambda_{i_j})$$

If $\lambda_{i_q} = \lambda_{i_r}$ for some q, r then the Vandermonde determinant of this term will be zero since its matrix will contain two identical rows. The only terms left contain all k distinct eigenvalues as roots. $\square$

Therefore, if we have a 1000×1000 matrix with just two distinct eigenvalues, one can just construct the 2×2 matrix T_2 to get both of them exactly. While matrices like this rarely occur in applications, the result is an interesting one.

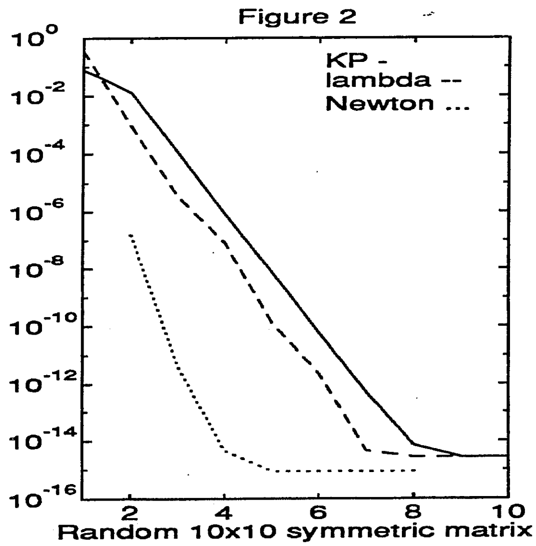
The problem at hand is to find error bounds or approximations for the eigenvalues of the matrices T_j. Cybenko's formula turns the problem into a search for the roots of polynomials. A well-known technique for finding such roots is Newton's method. The remainder of this paper will examine the application of one iteration of Newton's method to the characteristic polynomial $p_j(x)$ of T_j in order to approximate the eigenvalues of T_j.

Recall that Newton's method works as follows: given a function $f(x)$ and an initial guess x_0, then $x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}$ will be a closer approximation to the nearest root than x_0. Applying this first iteration to Cybenko's formula yields

$$x_1 = x_0 - \frac{\sum_{I \in \Delta nj} \gamma_{i_1}\cdots\gamma_{i_j} V^2(\lambda_{i_1}, \ldots, \lambda_{i_j}) \times (x_0 - \lambda_{i_1})\cdots(x_0 - \lambda_{i_j})}{\sum_{I \in \Delta nj} [\gamma_{i_1}\cdots\gamma_{i_j} V^2(\lambda_{i_1}, \ldots, \lambda_{i_j}) \sum_{I_p \in \Delta n(j-1)} (x_0 - \lambda_{i_{p_1}})\cdots(x_0 - \lambda_{i_{p_{j-1}}})]} \quad (10)$$

For the symmetric case, we can create an upper bound for T_j's largest eigenvalues θ_j by letting $x_0 = \lambda_n$. We know this is an upper bound since the characteristic polynomial $p(x)$ will have all real roots, so the concavity of p does not change for $x \geq \theta_j$, and we know $\lambda_n \geq \theta_j$ from the Kaniel-Paige theorem. Thus x_1 will be located between λ_n and θ_j. Figure

(2) shows that this bound is a much tighter one than the Kaniel-Paige bounds. In the unsymmetric case, the Newton formula yields just an approximation and not necessarily a bound since some of the eigenvalues may be complex. From Figure (3) one can see that this approximation is also fairly tight. These graphs indicate that the Newton estimate seems to have a lot of potential.



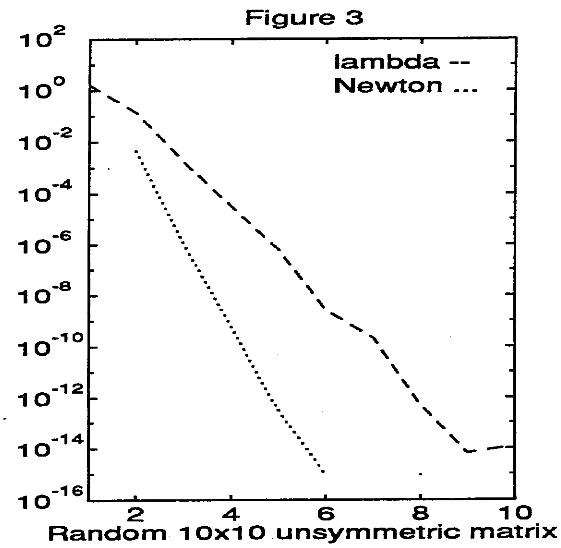


Figure (2) and Figure(3) plot the closeness of the bounds to the actual dominant eigenvalues of the T_j's for $j = 1:10$.

To be practical at all, this formula must be simplified so that it does not require prior knowledge of the entire eigenspace. Plugging in λ_n for x_0 causes many of the terms to cancel, but unfortunately this does not provide a simpler expression of equation (10). Some knowledge of the distribution of the eigenvalues can help this simplification.

Consider random matrices whose entries are uniformly distributed between 0 and 1. Such matrices can be generated on Matlab by the command $\mathbf{A} = \mathbf{rand(n)}$. A symmetric matrix can be generated from a random matrix A by $\mathbf{S} = (\mathbf{A} + \mathbf{A^T})/\mathbf{2}$. These matrices have a dominant eigenvalue at approximately $\frac{n}{2}$, and the rest are located within a circle of radius $\sqrt{\frac{n}{6}}$ about the origin (Silverstein [4]). To make things simpler, suppose the initial vector x has equal components in the direction of all the eigenvectors. Then we can cancel the γ_i's from equation (10). Actually, the choice of a random initial vector should have little impact on the speed of convergence of Lanczos. Regard Figure (4), which plots the convergence of the Lanczos eigenvalues to λ_n for a random initial vector and for one with equal components in the directions of the eigenvectors. The equal components vector gives a bad initial estimate[2] of $trace(A)/n$, but the convergence rates appear to be roughly the same. Since terms containing λ_n cancel, our Newton formula contains $\binom{n-1}{j}$ terms in the

[2] $x^T Ax = trace(A)/n$ in the symmetric case, where $x = \frac{1}{\sqrt{n}}V[1,\ldots,1]^T$. Obviously this guess is not even close to the dominant eigenvalue.

numerator containing j products of the form $\lambda_n - \lambda_i$. To maintain the upper bound, we can underestimate the difference by substituting $(\lambda_n - \sqrt{\frac{n}{6}})^j$. The denominator consists of $\binom{n}{j}$ terms containing $j-1$ products of the form $\lambda_n - \lambda_i$. We can overestimate by using $(\lambda_n + \sqrt{\frac{n}{6}})^{j-1}$. The Vandermonde determinants contain $\binom{j}{2}$ products of the form $\lambda_i - \lambda_j$. In the numerator these products do not include λ_n, although some in the denominator do. For this fraction, we can initially try $\frac{\sqrt{\frac{n}{6}}}{\lambda_n + \sqrt{\frac{n}{6}}}$. Our Newton approximation then becomes

$$
\begin{aligned}
x_1 &= \lambda_n - \frac{\binom{n-1}{j}}{\binom{n}{j}} \left(\frac{\sqrt{\frac{n}{6}}}{\lambda_n + \sqrt{\frac{n}{6}}} \right)^{2\binom{j}{2}} \frac{(\lambda_n - \sqrt{\frac{n}{6}})^j}{(\lambda_n + \sqrt{\frac{n}{6}})^{j-1}} \\
&= \lambda_n - \frac{n-j}{n} \left(\frac{\sqrt{\frac{n}{6}}}{\lambda_n + \sqrt{\frac{n}{6}}} \right)^{2\binom{j}{2}} \frac{(\lambda_n - \sqrt{\frac{n}{6}})^j}{(\lambda_n + \sqrt{\frac{n}{6}})^{j-1}} \qquad (11)
\end{aligned}
$$

Figure 4

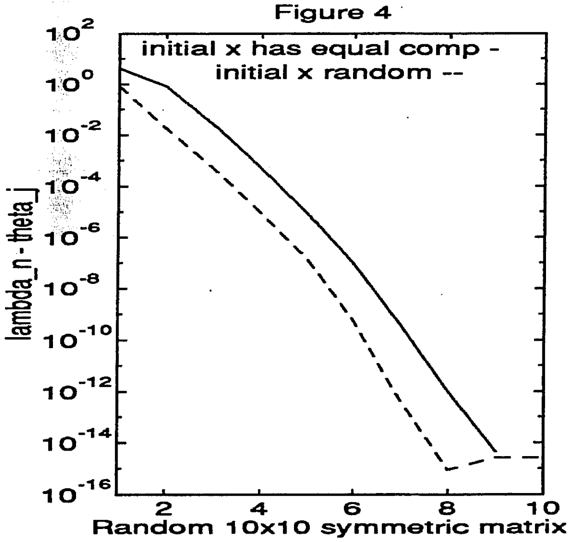


Figure 5

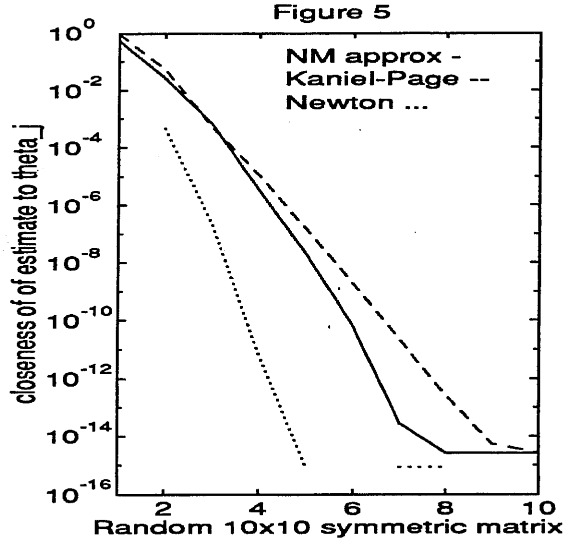


Figure (4) plots the convergence rate $\lambda_n - \theta_j$ for $j = 1 : 10$ for two starting vectors – one with equal components in the directions of all the eigenvectors and one random. Figure (5) plots the closeness of three different bounds to the dominant eigenvalues of the T_j's.

The exponent of $2\binom{j}{2}$ causes this fraction to approach zero very quickly, just as the Lanczos method locks in on the dominant eigenvalue very quickly. For large n, experimentation seems to show that this formula is still an upper bound for θ_j except for T_2. We can correct this by leaving out the term $\frac{(\lambda_n-\sqrt{\frac{n}{6}})^j}{(\lambda_n+\sqrt{\frac{n}{6}})^{j-1}}$, although we lose some initial tightness in doing so. Figure (5) plots the tightness of this approximation versus the Kaniel-Paige bound and the original Newton formula. Much of the tightness from the original Newton's method (before simplification) has been lost already by the estimations, so better estimates of the eigenvalue differences, particularly in the Vandermonde terms, should be an improvement.

Although our formula requires prior knowledge of the dominant eigenvalue, remember that the Kaniel-Paige bound uses three eigenvalues. These formulas can be used to predict the location of θ_j given λ_n. This may seem strange since the whole purpose of the Lanczos method is to find the θ_j's to approximate λ_n. Our Newton estimate is best used to give a better understanding of the Lanczos convergence and a general expectation of the accuracy of Lanczos.

This paper was written as a part of the Research Experience for Undergraduates at Oregon State University in the summer of 1993. The author would like to sincerely thank Dr. J.A.C. Weideman for his ideas and encouragement, without which this paper would not have been possible.

References

1 George Cybenko, An Explicit Formula for Lanczos Polynomials, *Linear Algebra and its Applications*, 88/89:99-115 (1987).

2 G. H. Golub and C. F. Van Loan, *Matrix Computations*, Johns Hopkins, Baltimore, 1983.

3 B.N. Parlett, D.R. Taylor, and Z.A. Liu, A Look-Ahead Lanczos Algorithm for Unsymmetric Matrices, *Math. Comp.* 44:105-124 (1985).

4 Jack Silverstein, private conversation, July 1993.

5 G. W. Stewart, *Introduction to Matrix Computations*, Academic Press, Orlando, 1973.