Estimating a Largest Eigenvector by Polynomial Algorithms with a Random Start

Columbia University Computer Science Department Report CUCS-023-96

Z. Leyk*
Institute for Scientific Computation
Texas A&M University

H. Woźniakowski†
Department of Computer Science, Columbia University
and
Institute of Applied Mathematics, University of Warsaw

May 20, 1996

Abstract

In [7] and [8], the power and Lanczos algorithms with random start for estimating the largest eigenvalue of an $n \times n$ large symmetric positive definite matrix were analyzed. In this paper we continue this study by estimating an eigenvector corresponding to the largest eigenvalue. We analyze polynomial algorithms using Krylov information for two error criteria: the randomized error and the randomized residual error.

For the randomized error, in contrast to [7] and [8], we prove that it is *not* possible to get distribution-free bounds. In fact, there exists a matrix for which *all* polynomial algorithms fail for approximating an eigenvector corresponding to the largest eigenvalue as long as the number of steps is less than n. This shows that the problem of estimating such an eigenvector is much harder than the problem of estimating the largest eigenvalue, and that even the random

*E-mail address: `zbigniew.leyk@isc.tamu.edu`
†Supported in part by the National Science Foundation and the Air Force Office of Scientific Research. This work was partly done while the author was visiting the Australian National University in 1994. E-mail address: `henryk@cs.columbia.edu`

start does not help. For the randomized error, the bounds must depend on the distribution of eigenvalues of the matrix. For the power algorithm, bounds depending on the ratio of the two largest eigenvalues are presented in [2]. We supply such bounds for the Lanczos algorithm.

For the randomized residual error, distribution-free bounds exist and we supply such bounds for the power and Lanczos algorithms. Asymptotic bounds, as well as bounds depending on the ratio of the two largest eigenvalues, are also presented.

1 Introduction

The eigenproblem for $n \times n$ large symmetric positive definite matrices A is usually solved by algorithms using Krylov information with an initial vector b. It is well known that if b is orthogonal to the eigenspace $V(A)$ of the largest eigenvalue then all algorithms must fail when approximating either the largest eigenvalue or a largest eigenvector. By a largest eigenvector we mean any normalized vector from $V(A)$.

This motivates the choice of a random vector b. Since usually we do not have any a priori information about the subspace $V(A)$, it seems natural to select the vector b randomly with uniform distribution over the n dimensional unit sphere. Computationally, this can be done as described, e.g., in [1] and [6].

This approach has been taken in [7] and [8] for estimating the largest eigenvalue. In this paper, we study the related problem of estimating a largest eigenvector. We consider polynomial algorithms that use Krylov information with a random start b. That is, we compute $u_k(b) = P(A)b$, where P is a polynomial of degree less than k.

We propose two criteria for measuring the error. The first one is the *randomized error*. This error is defined by taking the distance between the computed vector $u_k(b)$ and the eigenspace $V(A)$, and then by taking the expectation with respect to random vectors b. The second criterion is the *randomized residual error*. This is given by taking the (relative) residual $\|Au_k(b) - \lambda_1 u_k(b)\|/\lambda_1$ and the expectation with respect to b. Here λ_1 is the largest eigenvalue of A, and $\|\cdot\|$ is the Euclidean norm.

For the randomized error, we prove a negative result, see Theorem 1. Namely, there exists a symmetric positive definite $n \times n$ matrix A for which *all* polynomial algorithms fail to approximate a largest eigenvector as long as the number k of steps is less than n.

We comment on this troublesome matrix. Usually for each algorithm there exists a worst case matrix for which the algorithm behaves poorly. In our case, one matrix is bad for the whole class of polynomial algorithms. This troublesome matrix has the largest eigenvalue λ_1 of multiplicity one, with the rest of the eigenvalues pathologically close to λ_1. Then polynomial algorithms using Krylov information with a random start cannot distinguish between the eigenspace $V(A)$ and its orthogonal complement $V(A)^{\perp}$ well enough to compute a vector with a significantly large component along $V(A)$.

This negative result for estimating a largest eigenvector should be contrasted with the positive result for estimating the largest eigenvalue. For the latter problem, the power algorithm with a random start has a sharp bound roughly $\ln(n)/k$, whereas the Lanczos algorithm with a random start has a bound roughly $(\ln(n)/k)^2$, see [7]. Hence, the problem of estimating a largest eigenvector is much harder than the problem of estimating the largest eigenvalue, and even the random start does not help.

The negative result for the randomized error holds for $k < n$. For $k \geq n$, the Lanczos algorithm computes exactly a largest eigenvector with probability one. In fact, the number of steps required for the exact solution for the matrix A is equal to the total number of distinct eigenvalues of A.

For $k < n$, there are no distribution-free bounds for the randomized error of polynomial algorithms. There are, however, bounds dependent on the distribution of eigenvalues of the matrix A. Such bounds are provided for the power algorithm in [2] and they depend on the ratio ρ of the two largest eigenvalues and their multiplicities, see Theorem 2. We provide bounds

$$\frac{6.505}{1-\rho}\left(\frac{\ln(n)}{k-1}\right)^2 \qquad \text{and} \qquad \frac{1.61\sqrt[4]{n}}{1-\rho}\left(\frac{1-\sqrt{1-\rho}}{1+\sqrt{1-\rho}}\right)^{(k-1)/2}.$$

for the Lanczos algorithm in Theorem 5. Hence, if ρ is not too close to one, the randomized error of the Lanczos algorithm is small even for relatively small k.

We now turn to the randomized residual error. In this case, there exist distribution-free bounds. For the power algorithm, sharp bounds of order $\ln(n)/k$ are provided in Theorem 3. For the Lanczos algorithm, we are able to prove only upper bounds of order $(\ln(n)/k)^2$, see Theorem 4. As expected, this gives us one more reason to prefer the Lanczos algorithm over the power algorithm. The bounds presented are similar to bounds for the largest eigenvalue problem. We also provide asymptotic bounds, as well as bounds dependent on the ratio of the two largest eigenvalues and their multiplicities, for both the power and Lanczos algorithms.

In Section 6 we briefly report numerical tests. Since numerical tests for the power algorithm may be found in [2], we restrict ourselves to the Lanczos algorithm. We test the Lanczos algorithm for the tridiagonal matrix with 2's on the main diagonal and -1 on the two codiagonals. The tests seems to indicate that upper bounds of Theorems 4 and 5 may be improved by a factor of order $\ln(k\,n)$.

We conclude this introduction by a brief remark on related works. We have already mentioned the paper [2], where the same problem is studied for the power algorithm and for the randomized error. We stress that the randomized error in [2] is studied in the L_p-norm for arbitrary values of $p \in [1, +\infty]$. In particular, the results depend on a relation between multiplicities of the two largest eigenvalue and the parameter p. The results deteriorate for large p. In our paper we consider only the case $p = 2$.

The idea of using a random start for estimating eigenvectors by the power algorithm can be found in [15]. The author applies the power algorithm to the exponential matrix $\exp(-A)$ and analyzes convergence to a (not necessarily largest) eigenvector. In this case, the power algorithm is globally convergent and a random start is used to improve efficiency. Still even for $n = 2$, the problem may be hard, see [15].

Finally, we comment on the two papers [5] and [17]. In these two papers, the *average* case setting for estimating a largest eigenvector is analyzed. That is, it is assumed that matrices are distributed according to some probability measure, and the behaviour of algorithms is analyzed by taking the expectation with respect to matrices. In our case, we do not assume a distribution on matrices. In fact, we analyze the behaviour of (polynomial) algorithms for worst case matrices or for a fixed matrix. We consider random starting vectors, which corresponds to the randomized setting.

2 Definition of the Problem

Let A be an $n \times n$ large symmetric matrix. Without loss of generality and to simplify further notation we may assume that A is positive definite. Indeed, one can shift eigenvalues of A by adding a positive number β to the diagonal elements of A. If $\beta > \|A\|$ then after this shift the matrix is positive definite. Here one can choose an arbitrary norm, and the choice of $\|\cdot\|_1$ or $\|\cdot\|_\infty$ may be used to get an easy upper bound on the norm of A.

Let $\lambda_i = \lambda_i(A)$ denote the eigenvalues of the matrix A, $\lambda_1(A) \geq \lambda_2(A) \geq \ldots \geq \lambda_n(A) > 0$. By $v_i = v_i(A)$ we denote orthonormal eigenvectors corresponding to λ_i, $(v_i, v_j) = \delta_{ij}$, where $(\cdot,\cdot)$ is the Euclidean inner product of vectors. By $\|\cdot\|$ we denote the Euclidean norm of vectors.

We want to compute an approximation to a largest eigenvector, i.e., to an eigenvector corresponding to the largest eigenvalue $\lambda_1(A)$. More precisely, for a given (presumably small) positive number ε we want to compute a vector $u = u(A)$ of norm 1 such that the error between $u(A)$ and the eigenvectors corresponding to $\lambda_1(A)$ does not exceed ε, i.e.,

$$\inf_{v \in V(A)} \|u(A) - v\| \leq \varepsilon, \tag{1}$$

where

$$V(A) \ = \ \mathrm{span}\{v : Av = \lambda_1(A)v\}$$

is the eigenspace corresponding to the largest eigenvalue.

Obviously, if $\varepsilon \geq 1$ then an arbitrary $u(A)$ satisfies (1) since we can take $v = 0$. To avoid this trivial case, we assume that $\varepsilon \in [0, 1)$.

If n is large, say, of order 10^3 or more then it is prohibitively expensive to use well known algorithms such as QR or QL. Instead, it is reasonable to assume that the information about the matrix A is supplied by a subroutine that computes Az for any vector z. If A is sparse, which often is the case, the time and storage needed to perform the matrix-vector multiplication Az is proportional to n.

We therefore assume that *Krylov* information consisting of $k-1$ matrix-vector multiplications, $k \geq 1$,

$$N_k(A, b) = [b, Ab, \ldots, A^{k-1}b], \tag{2}$$

is used to compute the approximation $u(A)$. That is, $u(A) = \phi_k(N_k(A, b))$ for some mapping $\phi_k : \boldsymbol{R}^{nk} \mapsto \boldsymbol{R}$. Here, b is a nonzero vector which, without loss of generality, may be normalized such that $||b|| = 1$.

Krylov information can be written as $[z_1, z_2, \ldots, z_k]$ with $z_1 = b$ and $z_i = Az_{i-1}$. This shows that it can be computed in time of $k-1$ matrix-vector multiplications. The Krylov subspace is defined as

$$K_k = K(A, b, k) = \text{span}\{b, Ab, \ldots, A^{k-1}b\}.$$

In this paper we consider *polynomial* algorithms $u_k = u(A, b, k)$ that use Krylov information with $k-1$ matrix-vector multiplications. They are defined by assuming that $u_k \in K_k$, and has norm one. That is,

$$u(A, b, k) \;=\; c_0 b + c_1 Ab + \cdots + c_{k-1}A^{k-1}b \;=\; P(A)b,$$

where P is a polynomial of degree at most $k-1$, and $\|u_k\| = 1$.

Examples of polynomial algorithms include the power and (simple) Lanczos algorithms. For the power algorithm u^{pow} we have

$$u_k \;=\; u^{\text{pow}}(A, b, k) = \frac{Au_{k-1}}{\|Au_{k-1}\|} \quad \text{for } k = 1, 2, \ldots \tag{3}$$

with $u_0 = b$.

For the Lanczos algorithm u^{Lan} we have

$$u_k \;=\; u^{\text{Lan}}(A, b, k) \in K(A, b, k) \quad \text{for } k = 1, 2, \ldots$$

satisfying $\|u_k\| = 1$ and

$$R_k A u_k = \xi_k u_k. \tag{4}$$

Here, $R_k : \boldsymbol{R}^n \mapsto K_k$ is the orthogonal projector (matrix) onto K_k, i.e., $R_k v \in K_k$ and $(v - R_k v, w) = 0$ for any $v \in \boldsymbol{R}^n$ and $w \in K_k$. Moreover,

$$\xi_k = \max\{(Ax, x) : \|x\| = 1 \text{ and } x \in K_k\} \tag{5}$$

is the largest eigenvalue of $R_k A R_k$ and $u_0 = b$. By (4) and definition of ξ_k we get

$$(Au_k, u_k) = \max_{x \in K_k} \frac{(Ax, x)}{\|x\|^2}. \tag{6}$$

The analysis of convergence of the power algorithm is straightforward and may be found in most books on numerical analysis. The analysis of convergence of the Lanczos algorithm is more complex and some of it may be found in e.g., [3, 4, 10, 11, 12, 13, 14, 16].

Convergence of a polynomial algorithm $u(A, b, k)$ depends on distributions of eigenvalues of the matrix A and of the vector b. In particular, if b is orthogonal to the eigenspace $V(A)$ then *all* polynomial algorithms fail to converge to v_1. Indeed, in this case $u(A, b, k)$ is also orthogonal to $V(A)$ and its distance to the space $V(A)$ is 1. Hence, (1) does not hold for $\varepsilon < 1$.

Although it is impossible to guarantee that b is not orthogonal to the eigenspace $V(A)$, it is intuitively clear that this condition should hold for "almost all" vectors b. This is definitely the case if the vector b is chosen randomly, say, with uniform distribution μ on the n-dimensional sphere of radius one. The reader may consult [1] and [6, p. 130], where it is explained how such a vector can be generated computationally. Then, in fact, the vector b is *not* orthogonal to all eigenvectors $v_i(A)$ with probability 1.

In this paper we assume that the initial vectors b are chosen randomly. Let μ be a distribution of random vectors b. Since usually we do not have any a priori assumption about the location of largest eigenvectors, it is reasonable to choose μ in such a way that all directions of the space $\boldsymbol{R}^n$ are equally probable. This means that μ should be orthogonally invariant, i.e., $\mu(A) = \mu(QA)$ for any measurable set A and any orthogonal matrix Q. Furthermore, since the length of the random vector b does not matter, the measure μ should be concentrated on the unit sphere. This motivates our choice of μ as a *uniform* distribution over the unit sphere of $\boldsymbol{R}^n$, $\mu(\{b \in \boldsymbol{R}^n : ||b|| = 1\}) = 1$.

For any symmetric and positive definite matrix A, we select a *random* vector b according to the distribution μ. Then we compute Krylov information $N_k(A, b)$ and the approximation $u(b) = u(A, b, k)$ by a polynomial algorithm, e.g., by the power or Lanczos algorithm. Then

$$e^{\mathrm{ran}}(u, A, k) = \left(\int_{||b||=1} \inf_{v \in V(A)} \|u(b) - v\|^2 \, \mu(db) \right)^{1/2} \tag{7}$$

is the *randomized* error and

$$r^{\mathrm{ran}}(u, A, k) = \left(\int_{||b||=1} \frac{\|Au(b) - \lambda_1(A) u(b)\|^2}{\lambda_1^2(A)} \, \mu(db) \right)^{1/2} \tag{8}$$

is the *randomized residual* (relative) error.

3 Worst Distribution of Eigenvalues

In this section we consider the randomized error and show that there are no distribution-free bounds for polynomial algorithms. More precisely, we prove that there exists a symmetric positive matrix A for which *any* polynomial algorithm u_k with $k < n$ fails to approximate a largest eigenvector of A.

Let U_k denote the class of polynomial algorithms using Krylov information $N_k(A, b)$, and let $\mathcal{A}_n$ denote the class of $n \times n$ symmetric positive definite matrices.

Theorem 1

For $k < n$ we have

$$\sup_{A \in \mathcal{A}_n} \inf_{u \in U_k} e^{\mathrm{ran}}(u, A, k) \;=\; \sqrt{1 - \frac{1}{n - k + 1}} \;\geq\; \frac{1}{\sqrt{2}}.$$

For $k \geq n$, the Lanczos algorithm has zero randomized error,

$$e^{\mathrm{ran}}(u^{\mathrm{Lan}}, A, k) \;=\; 0, \quad \forall\, A \in \mathcal{A}_n.$$

Proof of Theorem 1

Assume that $k < n$. Let $b = \sum_{i=1}^n b_i v_i$, where v_i are orthonormal eigenvectors of A. Consider an arbitrary polynomial algorithm u from U_k. We have

$$u \;=\; \frac{\sum_{i=1}^n b_i \, P(\lambda_i) \, v_i}{\sqrt{\sum_{i=1}^n b_i^2 \, P^2(\lambda_i)}}, \tag{9}$$

where P belongs to the class $\mathcal{P}_k$ of polynomials of degree $\leq k - 1$. The error of the polynomial algorithm u is given by

$$\inf_{v \in V(A)} \|u - v\|^2 \;=\; \frac{\sum_{i=p+1}^n b_i^2 \, P^2(\lambda_i)}{\sum_{i=1}^n b_i^2 \, P^2(\lambda_i)},$$

where p is the multiplicity of the eigenvalue λ_1. Using a continuity argument we may restrict ourselves to polynomials P such that $P(\lambda_1) \neq 0$. Let $Q(t) = P(\lambda_1 t)/P(\lambda_1)$. Then $Q \in \mathcal{P}_k$ and $Q(1) = 1$. Let $\mathcal{P}_k(1)$ denote such polynomials. Thus, for $x_i = \lambda_i/\lambda_1 \in (0, 1]$ we have

$$\inf_{v \in V(A)} \|u - v\|^2 \;=\; \frac{\sum_{i=p+1}^n b_i^2 \, Q^2(x_i)}{b_1^2 + \sum_{i=2}^n b_i^2 \, Q^2(x_i)}. \tag{10}$$

Let $Q^* \in \mathcal{P}_k(1)$ be a polynomial for which

$$\min_{Q \in \mathcal{P}_k(1)} \frac{\sum_{i=p+1}^n b_i^2 \, Q^2(x_i)}{b_1^2 + \sum_{i=2}^n b_i^2 \, Q^2(x_i)} = \frac{\sum_{i=p+1}^n b_i^2 \, Q^*(x_i)^2}{b_1^2 + \sum_{i=2}^n b_i^2 \, Q^*(x_i)^2}\,.$$

Observe that Q^* depends on the vector b and the matrix A through its normalized eigenvalues x_i. Then for $u^* = P^*(A)b$ with $P^*(t) = Q^*(t/\lambda_1)$ we have

$$\inf_{v\in V(A)} \|u-v\|^2 \geq \inf_{v\in V(A)} \|u^*-v\|^2.$$

Hence, the polynomial algorithm u^* is best possible, and it is enough to find its randomized error,

$$e_k^2 = \int_{\|b\|=1} \frac{\sum_{i=p+1}^n b_i^2\, Q^*(x_i)^2}{b_1^2 + \sum_{i=2}^n b_i^2\, Q^*(x_i)^2}\, \mu(db). \tag{11}$$

Since we are interested in the worst possible matrix A, we take A with $p=1$. We want to show that

$$\sup_{A\in\mathcal{A}_n} \int_{\|b\|=1} \frac{\sum_{i=2}^n b_i^2\, Q^*(x_i)^2}{b_1^2 + \sum_{i=2}^n b_i^2\, Q^*(x_i)^2}\, \mu(db) \;=\; \int_{\|b\|=1} \frac{\sum_{i=2}^{n-k+1} b_i^2}{b_1^2 + \sum_{i=2}^{n-k+1} b_i^2}\, \mu(db). \tag{12}$$

From the above equality we directly obtain

$$\int_{\|b\|=1} \frac{\sum_{i=2}^{n-k+1} b_i^2}{b_1^2 + \sum_{i=2}^{n-k+1} b_i^2}\, \mu(db) \;=\; 1 \;-\; \int_{\|b\|=1} \frac{b_1^2}{\sum_{i=1}^{n-k+1} b_i^2}\, \mu(db) \;=\; 1 - \frac{1}{n-k+1}.$$

The last equality follows from the fact that $a_j = \int_{\|b\|=1} b_j^2/(\sum_{i=1}^{n-k+1} b_i^2)\mu(db)$ does not depend on j, and $\sum_{j=1}^{n-k+1} a_j = 1$. Thus $a_j = 1/(n-k+1)$.

Hence it is enough to prove (12). We show an upper bound for the left hand side of (12). Let $Q(x)=1$ for $k=1$ and

$$Q(x) = \frac{(x-x_n)\ldots(x-x_{n-k+2})}{(1-x_n)\ldots(1-x_{n-k+2})}$$

for $k\geq 2$. Notice that $Q\in\mathcal{P}_k(1)$. Also $Q(x_i)=0$ for $i=n-k+2,\ldots,n$ and $|Q(x_i)|\leq 1$ for $i=1,\ldots,n-k+1$ (due to $1=x_1\geq x_2\geq\ldots\geq x_n$). Since Q^* minimizes the integrand values, we get

$$\sup_{A\in\mathcal{A}_n} \int_{\|b\|=1} \frac{\sum_{i=2}^n b_i^2\, Q^*(x_i)^2}{b_1^2 + \sum_{i=2}^n b_i^2\, Q^*(x_i)^2}\, \mu(db) \leq \int_{\|b\|=1} \frac{\sum_{i=2}^{n-k+1} b_i^2}{b_1^2 + \sum_{i=2}^{n-k+1} b_i^2}\, \mu(db). \tag{13}$$

We now show a lower bound for the left hand side of (12). Since

$$\frac{\sum_{i=2}^n b_i^2\, Q^*(x_i)^2}{b_1^2 + \sum_{i=2}^n b_i^2\, Q^*(x_i)^2} = 1 - \frac{b_1^2}{b_1^2 + \sum_{i=2}^n b_i^2 Q^*(x_i)^2}$$

then

$$\sum_{i=2}^n b_i^2 Q^*(x_i)^2 = \min_{Q\in\mathcal{P}_k(1)} \sum_{i=2}^n b_i^2 Q^2(x_i). \tag{14}$$

Note that $Q(x) = 1$ belongs to $\mathcal{P}_k(1)$. Then the fact that $\sum_{i=1}^n b_i^2 = 1$ yields

$$\sum_{i=2}^{n} b_i^2 Q^*(x_i)^2 \leq 1. \tag{15}$$

Let $\delta \in (0, 1/n)$. Consider b_i satisfying $|b_i| \geq \delta$, $i = 1, \ldots, n$. Then by (15)

$$|Q^*(x_i)| \leq \frac{1}{\delta}, \quad \text{for any } i = 2, \ldots, n. \tag{16}$$

For fixed $x_{n-k+2}, \ldots, x_n$ and $x_{n+1} = 1$ we have

$$Q^*(x) = \sum_{i=0}^{k-1} Q^*(x_{n-i+1})\, l_i(x),$$

where

$$l_i(x) = \prod_{j=0, j\neq i}^{k-1} \frac{x - x_{n-j+1}}{x_{n-i+1} - x_{n-j+1}}, \quad i = 0, \ldots, k-1.$$

We want to estimate the first derivative of Q^*. Since

$$(Q^*(x))' = \sum_{i=0}^{k-1} Q^*(x_{n-i+1})\, l_i'(x)$$

then by (16)

$$|(Q^*(x))'| \leq \frac{1}{\delta} \sum_{i=1}^{k-1} |l_i'(x)| \leq C, \quad \text{for any } x \in [0, 1], \tag{17}$$

where $C = C(\delta, k, x_{n-k+2}, \ldots, x_n)$ but C does not depend on $x_2, \ldots, x_{n-k+1}$ and $b_1, \ldots, b_n$ (due to $|b_i| \geq \delta$).

Now we take matrices A such that their normalized eigenvalues x_i for $i \in [2, n-k+1]$, tend to 1^-. That is, we take $\eta \in (0, 1/C)$ such that $|1 - x_j| \leq \eta$ for $j = 2, \ldots, n-k+1$. We have

$$Q^*(x_j) = Q^*(1) + (Q^*(\xi_j))'(x_j - 1), \quad \text{for } \xi_j \in [0, 1].$$

Since $Q^*(1) = 1$ then by (17)

$$|1 - Q^*(x_j)| \leq C|1 - x_j| \leq C\eta$$

and

$$Q^*(x_j) \geq 1 - C\eta > 0, \quad \text{for } j = 2, \ldots, n-k+1. \tag{18}$$

By (18) it follows that

$$\sum_{i=2}^{n} b_i^2 Q^*(x_i)^2 \geq \sum_{j=2}^{n-k+1} b_j^2 Q^*(x_j)^2 \geq (1 - C\eta)^2 \sum_{j=2}^{n-k+1} b_j^2.$$

We get

$$\begin{aligned}
&\int_{\|b\|=1} \frac{\sum_{i=2}^n b_i^2\, Q^*(x_i)^2}{b_1^2+\sum_{i=2}^n b_i^2\, Q^*(x_i)^2}\,\mu(db) \\
\geq\; &\int_{\|b\|=1,|b_i|\geq\delta} \left(1-\frac{b_1^2}{b_1^2+\sum_{i=2}^n b_i^2 Q^*(x_i)^2}\right)\,\mu(db) \\
\geq\; &\int_{\|b\|=1,|b_i|\geq\delta} \left(1-\frac{b_1^2}{b_1^2+(1-C\eta)^2\sum_{i=2}^{n-k+1} b_i^2}\right)\,\mu(db).
\end{aligned}$$

Since C does not depend on η, letting $\eta\to 0^+$ we obtain

$$\begin{aligned}
\sup_{A\in\mathcal{A}_n}\int_{\|b\|=1} \frac{\sum_{i=2}^n b_i^2\, Q^*(x_i)^2}{b_1^2+\sum_{i=2}^n b_i^2\, Q^*(x_i)^2} \qquad &\mu(db)\geq \\
&\int_{\|b\|=1,|b_i|\geq\delta}\left(1-\frac{b_1^2}{b_1^2+\sum_{i=2}^{n-k+1} b_i^2}\right)\,\mu(db).
\end{aligned}$$

Letting $\delta\to 0^+$ we find that

$$\begin{aligned}
&\sup_{A\in\mathcal{A}_n}\int_{\|b\|=1} \frac{\sum_{i=2}^n b_i^2\, Q^*(x_i)^2}{b_1^2+\sum_{i=2}^n b_i^2\, Q^*(x_i)^2}\,\mu(db) \\
\geq\; &\int_{\|b\|=1}\left(1-\frac{b_1^2}{b_1^2+\sum_{i=2}^{n-k+1} b_i^2}\right)\,\mu(db)=\int_{\|b\|=1}\frac{\sum_{i=2}^{n-k+1} b_i^2}{b_1^2+\sum_{i=2}^{n-k+1} b_i^2}\,\mu(db).
\end{aligned}$$

This completes the proof of the first part of the theorem.

The proof for $k\geq n$ is given in the proof of Theorem 5. $\square$

4 Power Algorithm

In this section we present bounds for the power algorithm. The randomized error of the power algorithm is analyzed in [2] in the L_p-norm with arbitrary $p\in[1,+\infty]$. For completeness, we recall the results for $p=2$.

Theorem 2 [2] *Let u^{pow} be the power algorithm defined by (3).*

(a) For any k, we have

$$\sup_{A\in\mathcal{A}_n} e^{\mathrm{ran}}(u^{\mathrm{pow}},A,k)=\sqrt{1-\frac{1}{n}}.$$

(b) For any symmetric positive definite matrix A, let p, $p < n$, and q denote the multiplicities of the two largest eigenvalues λ_1 and λ_{p+1}. Then

$$
\begin{aligned}
\lim_{k\to+\infty} \frac{e^{\mathrm{ran}}(u^{\mathrm{pow}}, A, k)}{(\lambda_{p+1}/\lambda_1)^k} &= \sqrt{\frac{q}{p-2}} && \text{for } p \ge 3, \\
\lim_{k\to+\infty} \frac{e^{\mathrm{ran}}(u^{\mathrm{pow}}, A, k)}{\sqrt{k}\,(\lambda_3/\lambda_1)^k} &= \sqrt{q\,\ln\frac{\lambda_1}{\lambda_3}} && \text{for } p = 2, \\
\lim_{k\to+\infty} \frac{e^{\mathrm{ran}}(u^{\mathrm{pow}}, A, k)}{(\lambda_2/\lambda_1)^{k/2}} &= \sqrt{\sqrt{\pi}\,\frac{\Gamma((q+1)/2)}{\Gamma(q/2)}} && \text{for } p = 1.
\end{aligned}
$$

We now present bounds for the randomized residual error. To simplify some estimates we assume that $n \ge 8$.

Theorem 3 *Let u^{pow} be the power algorithm defined by (3).*

(a) For any symmetric positive definite matrix A and for any $k \ge 1$ we have

$$r^{\mathrm{ran}}(u^{\mathrm{pow}}, A, k) \le \frac{1.12\,\ln(n)}{k}.$$

(b) For any $k > 1 + \frac{1}{2}\ln(n/\ln(n))$, let A be any symmetric matrix with exactly two distinct eigenvalues $\lambda_1 > 0$ and $\lambda_i = \lambda_1(1 - \frac{1}{2}\ln(n/\ln(n))/k)$, for $i = 2, 3, \ldots, n$. Then for large n and k,

$$r^{\mathrm{ran}}(u^{\mathrm{pow}}, A, k) \ge 0.5\,\frac{\ln(n)}{k}\,(1 + o(1)).$$

(c) For any symmetric positive definite matrix A, let p, $p < n$, and q denote the multiplicities of the two largest eigenvalues λ_1 and λ_{p+1}. Then

$$
\begin{aligned}
\lim_{k\to+\infty} \frac{r^{\mathrm{ran}}(u^{\mathrm{pow}}, A, k)}{(\lambda_{p+1}/\lambda_1)^k} &= \sqrt{\frac{q}{p-2}}\left(1 - \frac{\lambda_{p+1}}{\lambda_1}\right) && \text{for } p \ge 3, \\
\lim_{k\to+\infty} \frac{r^{\mathrm{ran}}(u^{\mathrm{pow}}, A, k)}{\sqrt{k}\,(\lambda_3/\lambda_1)^k} &= \sqrt{q}\left(1 - \frac{\lambda_3}{\lambda_1}\right)\sqrt{\ln\frac{\lambda_1}{\lambda_3}} && \text{for } p = 2, \\
\lim_{k\to+\infty} \frac{r^{\mathrm{ran}}(u^{\mathrm{pow}}, A, k)}{(\lambda_2/\lambda_1)^{k/2}} &= \sqrt{\sqrt{\pi}\,\frac{\Gamma((q+1)/2)}{\Gamma(q/2)}}\left(1 - \frac{\lambda_2}{\lambda_1}\right) && \text{for } p = 1.
\end{aligned}
$$

Observe that part (c) of Theorem 3 differs from part (b) of Theorem 2 only by the factor $1 - \lambda_{p+1}/\lambda_1$. Hence, the asymptotic behaviour of the power method is essentially the same for both the randomized error and the randomized residual error.

Proof of Theorem 3

Let $b = \sum_{i=1}^n b_i \, v_i$. From (3) we get

$$u_k = u^{\text{pow}}(A,b,k) = \frac{\sum_{i=1}^n b_i \, \lambda_i^k \, v_i}{\sqrt{\sum_{i=1}^n b_i^2 \, \lambda_i^{2k}}}. \tag{19}$$

As in Section 3, let $x_i = \lambda_i/\lambda_1 \in (0,1]$. Then by (19)

$$\frac{\|Au_k - \lambda_1 u_k\|^2}{\lambda_1^2} = \frac{\sum_{i=2}^n b_i^2 \, x_i^{2k}(1-x_i)^2}{b_1^2 + \sum_{i=2}^n b_i^2 \, x_i^{2k}}.$$

From this the randomized residual error of the power algorithm is given by

$$r_k^2 = r^{\text{ran}}(u^{\text{pow}}, A, k)^2 = \int_{\|b\|=1} \frac{\sum_{i=2}^n b_i^2 \, x_i^{2k}(1-x_i)^2}{b_1^2 + \sum_{i=2}^n b_i^2 \, x_i^{2k}} \, \mu(db). \tag{20}$$

We remark that for the largest eigenvalue problem, the randomized error of the power algorithm is similar to (20). The only difference is that the factor $(1-x_i)^2$ is replaced by $1-x_i$. Repeating step by step the proof of Theorem 3.1 of [7] with the extra factor $1-x_i$, we get

$$r_k^2 \,\le\, 2.064\,(1-\beta)^2 \sqrt{\frac{\pi\, n}{8}\,\beta^{2k} \,+\, \frac{1}{n}}, \quad \beta \in [0, \beta^*], \tag{21}$$

where $\beta^* = 1 - 2/(k+2)$.

Note that for $k \le 1.12\,\ln(n)$, part (a) of Theorem 3 trivially holds since

$$r_k^2 \le 1 \le (1.12\,\ln(n)/k)^2.$$

Assume thus that $k > 1.12\,\ln(n)$. Take now $\beta = 1 - \ln(n)/k$. Then $\beta \in (0, \beta^*]$ for $n \ge 8$. Obviously $\beta^{2k} \le 1/n^2$. Thus, we have

$$r_k^2 \le 2.064 \sqrt{\frac{\pi}{8n} \,+\, \frac{1}{\pi}} \left(\frac{\ln(n)}{k}\right)^2 \,\le\, \left(1.12\,\frac{\ln(n)}{k}\right)^2.$$

This proves part (a).

Part (b) of Theorem 3 follows easily from the proof of part (b) of Theorem 3.1 in [7].

We proceed to prove part (c) of Theorem 3. Recall that p and q are multiplicities of the two largest distinct eigenvalues of A. From (20) we can write as $k \to +\infty$,

$$r_k^2 \,=\, (1-x_{p+1}) \int_{\|b\|=1} \frac{x_{p+1}^{2k}(1-x_{p+1}) \sum_{i=p+1}^{p+q} b_i^2}{\sum_{i=1}^p b_i^2 + x_{p+1}^{2k} \sum_{i=p+1}^{p+q} b_i^2} \, \mu(db) \,\, (1+o(1)).$$

The integral above is analyzed in the proof of part (c) of Theorem 3.1 in [7]. Applying this analysis we obtain the asymptotic bounds of part (c). This completes the proof of Theorem 3. $\square$

5 Lanczos Algorithm

We now proceed to the Lanczos algorithm. The analysis of this algorithm is much more complex and we are able to present only upper bounds. For simplicity we assume that $k \geq 4$ and $n \geq 8$. It is convenient to start with the randomized residual error.

Theorem 4 *Let u^{Lan} be the Lanczos algorithm defined by (4).*
(a) For any symmetric positive definite matrix A, let m denote the number of distinct eigenvalues of A. Then
for $k \geq m$,

$$r^{\mathrm{ran}}(u^{\mathrm{Lan}}, A, k) = 0,$$

for $k \in [4, m-1]$,

$$r^{\mathrm{ran}}(u^{\mathrm{Lan}}, A, k) \leq 0.0803 \left(\frac{\ln(n(k-1)^8)}{k-1} \right)^2 \leq 6.505 \left(\frac{\ln(n)}{k-1} \right)^2.$$

(b) For any symmetric positive definite matrix A, let $p < n$ be the multiplicity of the largest eigenvalue λ_1, and let λ_{p+1} and λ_n be the second largest and the smallest eigenvalue of A. Then

$$r^{\mathrm{ran}}(u^{\mathrm{Lan}}, A, k) \leq 1.61 \sqrt[4]{n} \left(\frac{1 - \sqrt{(\lambda_1 - \lambda_{p+1})/(\lambda_1 - \lambda_n)}}{1 + \sqrt{(\lambda_1 - \lambda_{p+1})/(\lambda_1 - \lambda_n)}} \right)^{(k-1)/2}.$$

Proof of Theorem 4
For the Lanczos algorithm we have

$$u_k = u^{\mathrm{Lan}}(A, b, k) = \frac{\sum_{i=1}^n b_i P^*(\lambda_i) v_i}{\sqrt{\sum_{i=1}^n b_i^2 P^*(\lambda_i)^2}}, \tag{22}$$

where by (6) the polynomial P^* is of degree at most $k-1$ and satisfies

$$\frac{\sum_{i=1}^n b_i^2 \lambda_i P^*(\lambda_i)^2}{\sum_{i=1}^n b_i^2 P^*(\lambda_i)^2} = \max_{P \in \mathcal{P}_k} \frac{\sum_{i=1}^n b_i^2 \lambda_i P^2(\lambda_i)}{\sum_{i=1}^n b_i^2 P^2(\lambda_i)}.$$

As before, let $x_i = \lambda_i / \lambda_1$ and $Q^*(t) = P^*(\lambda_1 t)/P^*(\lambda_1)$. Then the error of the Lanczos algorithm is given by, see (10),

$$\inf_{v \in V} \|u_k - v\|^2 = \frac{\sum_{i=p+1}^n b_i^2 Q^*(x_i)^2}{b_1^2 + \sum_{i=2}^n b_i^2 Q^*(x_i)^2}. \tag{23}$$

Since

$$\frac{\lambda_1 - \xi_k}{\lambda_1} = \min_{Q \in \mathcal{P}_k(1)} \frac{\sum_{i=p+1}^n b_i^2 Q^2(x_i)(1-x_i)}{b_1^2 + \sum_{i=2}^n b_i^2 Q^2(x_i)} \tag{24}$$

where ξ_k is defined in (5), the polynomials $Q^* \in \mathcal{P}_k(1)$ satisfy

$$\frac{\sum_{i=p+1}^n b_i^2\, Q^*(x_i)^2 (1-x_i)}{b_1^2 + \sum_{i=2}^n b_i^2\, Q^*(x_i)^2} = \min_{Q \in \mathcal{P}_k(1)} \frac{\sum_{i=p+1}^n b_i^2\, Q^2(x_i)(1-x_i)}{b_1^2 + \sum_{i=2}^n b_i^2\, Q^2(x_i)}$$

The residuals of the Lanczos algorithm are given by

$$\frac{Au_k - \lambda_1 u_k}{\lambda_1} = \frac{\sum_{i=p+1}^n b_i\, Q^*(x_i)\,(x_i - 1) v_i}{\left(\sum_{i=1}^n b_i^2\, Q^*(x_i)^2\right)^{1/2}}.$$

Hence

$$\frac{\|Au_k - \lambda_1 u_k\|^2}{\lambda_1^2} = \frac{\sum_{i=p+1}^n b_i^2\, Q^*(x_i)^2\,(1-x_i)^2}{b_1^2 + \sum_{i=2}^n b_i^2\, Q^*(x_i)^2}$$

and the randomized error of the Lanczos algorithm can be estimated as

$$\begin{aligned} r_k^2 = r^{\mathrm{ran}}(u^{\mathrm{Lan}}, A, k)^2 &= \int_{\|b\|=1} \frac{\sum_{i=p+1}^n b_i^2\, Q^*(x_i)^2\,(1-x_i)^2}{b_1^2 + \sum_{i=2}^n b_i^2\, Q^*(x_i)^2}\, \mu(db) \\ &= \int_{\|b\|=1} \min_{Q \in \mathcal{P}_k(1)} \frac{\sum_{i=p+1}^n b_i^2\, Q^2(x_i)\,(1-x_i)^2}{b_1^2 + \sum_{i=2}^n b_i^2\, Q^2(x_i)}\, \mu(db) \\ &\le \min_{Q \in \mathcal{P}_k(1)} \int_{\|b\|=1} \frac{\sum_{i=p+1}^n b_i^2\, Q^2(x_i)\,(1-x_i)^2}{b_1^2 + \sum_{i=2}^n b_i^2\, Q^2(x_i)}\, \mu(db). \end{aligned}$$

We now prove part (a). Assume that $k \ge m$. Then we can set $Q(x_i) = 0$ for $i = p+1, p+2, \dots, n$. Since all these points x_i are different from one, and we have at most $m-1$ equations, we can choose Q such that $Q \in \mathcal{P}_k(1)$. Then $r_k = 0$, as claimed.

Assume now that $k \le m-1$. To estimate r_k, we note that the error of the Lanczos algorithm for the largest eigenvalue problem is given by a formula similar to the formula for r_k. The only difference is that instead of the factor $(1-x_i)^2$ in the above formulas we have $1 - x_i$. Therefore the major part of the proof of Theorem 3.2 from [7] can be also applied for the largest eigenvalue problem. This leads to the following estimate on r_k^2,

$$r_k^2 \le 0.824 \sqrt{\pi^2\, n\, e^{-4(k-1)\gamma} + 2\gamma^8}, \qquad \forall\, \gamma \in [0,1]. \tag{25}$$

If $k - 1 \le \sqrt{0.0803}\, \ln\left(n(k-1)^8\right)$ then part (a) is trivial since $r_k \le 1$.

For $k - 1 > \sqrt{0.0803}\, \ln\left(n(k-1)^8\right)$ take

$$\gamma = \frac{1}{4(k-1)} \ln \frac{2^{15}\pi^2 n\,(k-1)^8}{\left(\ln(n\,(k-1)^8)\right)^8}.$$

It is easy to check that $\gamma > 0$. For $n \geq 8$ and $k \geq 4$, we have $2^{15}\pi^2 \leq \left(\ln(n\,(k-1)^8)\right)^8$ and

$$\gamma \;\leq\; \frac{1}{4(k-1)}\,\ln(n(k-1)^8) \;\leq\; 1.$$

Substituting γ to (25) we get

$$r_k^2 \;\leq\; \frac{0.824}{128}\left(\frac{\ln(n(k-1)^8)}{k-1}\right)^4.$$

This leads to the estimate of part (a). Part (b) directly follows from the proof of part (b) of Theorem 3.2 from [7]. □

We now present upper bounds for the randomized error of the Lanczos algorithm. In view of Theorem 1, these bounds must depend on the eigenvalues of the matrix.

Theorem 5 *Let $u^{\rm Lan}$ be the Lanczos algorithm defined by (4). For any symmetric positive definite matrix A, let m denote the number of distinct eigenvalues, let $p < n$ be the multiplicity of the largest eigenvalue λ_1, and let λ_{p+1} and λ_n be the second largest and the smallest eigenvalue of A. Then for $k \geq m$,*

$$e^{\rm ran}(u^{\rm Lan}, A, k) \;=\; 0,$$

and for $k \in [4, m-1]$,

$$\begin{aligned} e^{\rm ran}(u^{\rm Lan}, A, k) \;&\leq\; 0.0803\,\frac{\lambda_1}{\lambda_1-\lambda_{p+1}}\left(\frac{\ln(n(k-1)^8)}{k-1}\right)^2 \\ &\leq\; 6.505\,\frac{\lambda_1}{\lambda_1-\lambda_{p+1}}\left(\frac{\ln(n)}{k-1}\right)^2, \end{aligned}$$

$$e^{\rm ran}(u^{\rm Lan}, A, k) \leq 1.61\,\sqrt[4]{n}\,\frac{\lambda_1}{\lambda_1-\lambda_{p+1}}\left(\frac{1-\sqrt{(\lambda_1-\lambda_{p+1})/(\lambda_1-\lambda_n)}}{1+\sqrt{(\lambda_1-\lambda_{p+1})/(\lambda_1-\lambda_n)}}\right)^{(k-1)/2}.$$

Proof of Theorem 5

The case $k \geq m$ follows easily from Theorem 4. For $k \leq m-1$, we use (23) and conclude that the randomized error e_k of the Lanczos algorithm is given by

$$\begin{aligned} e_k^2 \;&=\; \int_{\|b\|=1} \frac{\sum_{i=p+1}^n b_i^2\,Q^*(x_i)^2(1-x_i)^2\,\frac{1}{(1-x_i)^2}}{b_1^2+\sum_{i=2}^n b_i^2\,Q^*(x_i)^2}\,\mu(db) \\ &\leq\; \frac{1}{(1-x_{p+1})^2}\int_{\|b\|=1} \frac{\sum_{i=p+1}^n b_i^2\,Q^*(x_i)^2(1-x_i)^2}{b_1^2+\sum_{i=2}^n b_i^2\,Q^*(x_i)^2}\,\mu(db) \\ &=\; \frac{1}{(1-x_{p+1})^2}\int_{\|b\|=1} \min_{Q\in\mathcal{P}_k(1)} \frac{\sum_{i=p+1}^n b_i^2\,Q^2(x_i)(1-x_i)}{b_1^2+\sum_{i=2}^n b_i^2\,Q^2(x_i)}\,\mu(db). \end{aligned}$$

The last integral is equal to the square of the randomized residual error r_k^2, see the proof of Theorem 4. Hence,

$$e_k \leq \frac{1}{1 - x_{p+1}} r_k,$$

and all estimates of Theorem 5 follow from the estimates of r_k in Theorem 4. $\square$

6 Numerical Tests

We tested several matrices with many pseudo-random starting vectors b. Without loss of generality, see [7], we restricted ourselves only to symmetric tridiagonal matrices. Vectors b were uniformly distributed over the unit sphere of $\boldsymbol{R}^n$, and they were generated in the same way as in [7]. The tests were performed on a Sun-4/75 workstation with the round-off unit of order 10^{-16}. All calculations were done in double precision using the numerical package Meschach [9].

The purpose of the numerical tests was, in particular, to verify the sharpness of the bounds of Theorem 4 and 5. We report our results for the tridiagonal matrix A of dimension $n = 250$,

$$A = \begin{pmatrix} 2 & -1 & 0 & 0 & \dots & 0 \\ -1 & 2 & -1 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & -1 & 2 & -1 \\ 0 & \dots & 0 & 0 & -1 & 2 \end{pmatrix}$$

with eigenvalues

$$\lambda_{n+1-k} = 4 \sin^2 \frac{k\pi}{2(n+1)}, \quad k = 1, \dots, n,$$

and normalized eigenvectors

$$v_{n+1-k} = \sqrt{\frac{2}{n+1}} \left[\sin \frac{ki\pi}{n+1} \right]_{i=1}^{n}.$$

For each pseudo-random vector b we performed the Lanczos algorithm for $k = 1, 2, \dots$ as long as the errors $|\lambda_1(A) - \xi^{\mathrm{Lan}}(A, b, k)|/\lambda_1(A)$ were greater than 0.001. This termination criterion allowed us to compare our results with those reported in [7]. For all tested b and k we obtained

$$5.5 \cdot 10^{-5} \;\leq\; \inf_{v \in V} \|u(A,b,k) - v\| \left(\frac{k}{\ln(n)}\right)^2 \left(1 - \frac{\lambda_2}{\lambda_1}\right) \;\leq\; 0.0077,$$

$$0.032 \;\leq\; \frac{\|Au(A,b,k) - \lambda_1(A)\, u(A,b,k)\|}{\lambda_1(A)} \left(\frac{k}{\ln(n)}\right)^2 \;\leq\; 0.26\,.$$

In Table 1 we report the randomized errors achieved after $k-1$ steps of the Lanczos algorithm for ten different values of k which are listed in the first column. The second column contains the randomized errors defined as

$$e^{\text{ran}} = \left(\frac{1}{30} \sum_{i=1}^{30} \inf_{v \in V} \|u(A,b_i,k) - v\|^2\right)^{1/2},$$

where b_i is the ith pseudo-random vector. The third column presents upper bounds on the randomized errors from Theorem 5, i.e.,

$$e^{\text{up}} = 0.0803 \,\frac{\lambda_1}{\lambda_1 - \lambda_2} \left(\frac{\ln(n(k-1)^8)}{k-1}\right)^2.$$

We compute the ratios between the observed errors and their upper bounds in the fourth column, $r_1 = e^{\text{up}}/e^{\text{ran}}$. The fifth column displays the randomized residual errors after $k-1$ steps of the Lanczos algorithm defined as

$$r^{\text{ran}} = \left(\frac{1}{30} \sum_{i=1}^{30} \frac{\|Au(A,b,k) - \lambda_1(A)u(A,b,k)\|^2}{\lambda_1^2(A)}\right)^{1/2}$$

The next column shows upper bounds on the randomized residual errors from Theorem 4, i.e.,

$$r^{\text{up}} = 0.0803 \left(\frac{\ln(n(k-1)^8)}{k-1}\right)^2.$$

The last column contains the ratios between the observed residual errors and their upper bounds, $r_2 = r^{\text{up}}/r^{\text{ran}}$.

In Table 2 and 3 we indicate how many steps were needed to achieve randomized error and randomized residual error no greater than ε for six different values of ε. The values of ε are displayed in the first row of the tables. The second row of the tables shows the randomized number k^{ran} and l^{ran} of performed steps with $k^{\text{ran}} = \sum_{i=1}^{30} k(A,b_i)/30$ and $l^{\text{ran}} = \sum_{i=1}^{30} l(A,b_i)/30$, where $k(A,b_i)$ and $l(A,b_i)$ were the number of steps needed for the pseudo-random vector b_i for randomized error and randomized residual error respectively. The third row gives the minimal $k = k^{\text{up}}$ and $l = l^{\text{up}}$ such that

$$0.0803 \,\frac{\lambda_1}{\lambda_1 - \lambda_2} \left(\frac{\ln(n(k-1)^8)}{k-1}\right)^2 \;\leq\; \varepsilon \tag{26}$$

$k-1$	e^{ran}	e^{up}	r_1	r^{ran}	r^{up}	r_2
10	0.96	3920	4080	0.016	0.46	28.6
20	0.91	1490	1640	0.0059	0.175	29.4
30	0.86	814	946	0.0037	0.096	26.2
40	0.80	524	659	0.0023	0.062	27.2
50	0.76	371	487	0.0016	0.044	28.1
60	0.73	278	379	0.0013	0.033	24.3
70	0.72	218	303	0.001	0.026	25.5
80	0.65	176	270	0.00077	0.021	26.8
90	0.69	145	210	0.00067	0.017	25.5
100	0.64	123	192	0.00061	0.014	23.6

Table 1: Average errors for the Lanczos algorithm

and

$$0.0803 \left(\frac{\ln(n(l-1)^8)}{l-1} \right)^2 \leq \varepsilon, \tag{27}$$

which are taken from theoretical bounds for the Lanczos algorithm, see Theorem 4 and 5. The fourth row presents the ratios between the previous two numbers, $r_1 = k^{\text{up}}/k^{\text{ran}}$ and $r_2 = l^{\text{up}}/l^{\text{ran}}$.

ε	0.5	0.1	0.05	0.025	0.02	0.01
k^{ran}	126.2	175.5	190.7	200.7	197.0	219.9
k^{up}	2522	6237	9182	13494	15269	22394
r_1	20.0	35.5	48.1	67.2	77.5	101.8

Table 2: Average number of steps to satisfy
(26)

As we see the theoretical bound exceeds the actual value by a factor of at most 20 for errors and at least 7.3 for residuals. This indicates once more that the factor $\ln(n(k-1)^8)$ may be an overestimate in the theoretical bound. Observe also that all k^{up}'s and some l^{up}'s are greater than the dimension $n = 250$.

ε	0.05	0.01	0.005	0.0025	0.001	$5.0_{10}-4$
l^{ran}	5.8	15.6	24.0	36.2	65.6	101.3
l^{up}	47.0	127.0	192.0	290.0	496.0	741.0
r_2	8.2	8.1	8.0	8.0	7.6	7.3

Table 3: Average number of steps to satisfy (27)

References

[1] G. E. P. Box and M. E. Muller. A Note on the Generation of Random Normal Deviates. *Ann. Math. Statist.* **29** (1958), 610–611.

[2] Gianna M. Del Corso. Estimating an Eigenvector by the Power Method with a Random Start. Submitted for publication, 1995.

[3] W. Kahan and B. N. Parlett. How Far Should We Go with the Lanczos Process? in *Sparse Matrix Computations* eds. J. Bunch and D. Rose, Academic Press, New York, (1976), 131–144.

[4] S. Kaniel. Estimates for Some Computational Techniques in Linear Algebra. *Math. Comp.* **20** (1966), 369–378.

[5] E. Kostlan. Statistical complexity of dominant eigenvector calculation. *J. of Complexity* **7** (1991),. 371–379.

[6] D. E. Knuth. *The Art of Computer Science, Vol. 2: Seminumerical Algorithms.* 2nd ed. Addison-Wesley, Reading, MA., 1981.

[7] J. Kuczyński and H. Woźniakowski. Estimating the largest eigenvalue by the power and Lanczos algorithms with a random start. *SIAM J. Matrix Anal. Appl.* 13 (1992),. 1094–1122.

[8] J. Kuczyński and H. Woźniakowski. Probabilistic bounds on the extremal eigenvalues and condition number by the Lanczos algorithm. *SIAM J. Matrix Anal. Appl.* 15 (1994), 672–691.

[9] Z. Leyk and D. E. Stewart. Meschach: Matrix Computation in C. Proceedings of the CMA vol. **32**, Australian National University, 1994, pp. 1–240.

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

[11] C. C. Paige. Computational Variants of the Lanczos Method for the Eigenproblem. *J. Inst. Math. Appl.* **10** (1972), 373–381.

[12] B. N. Parlett. *The Symmetric Eigenvalue Problem.* Prentice-Hall, Inc., Englewood Cliffs, N. J., 1980.

[13] Y. Saad. On the Rates of Convergence of the Lanczos and the Block Lanczos Methods. *SIAM J. Numer. Anal.* **17** (1980), 687–706.

[14] D. S. Scott. *Analysis of the Symmetric Lanczos Process.* Ph. D. Thesis, University of California at Berkeley, Berkeley, CA., Memorandum NCB/ERLM 78/40, 1978.

[15] M. Shub. The geometry and topology of dynamical systems and algorithms for numerical problems. In *Proc. of the 1983 Beijing Symposium on Differential Geometry and Differential Equations*, Ed. Liao Shantao. Science Press, Beijing, China, 1986.

[16] J. H. Wilkinson. *The Algebraic Eigenvalue Problem.* Oxford Univ. Press, London and New York, 1965.

[17] P. E. Wright. Statistical complexity of the power method for Markov chains. *J. of Complexity* **5** (1989), 119–143.