Abstract. The EISPACK routine TINVIT is an implementation of inverse iteration for computing eigenvectors of real symmetric tridiagonal matrices. Experiments have shown that the eigenvectors computed with TINVIT are numerically less accurate than those from implementations of Cuppen's divide and conquer method (TREEQL) and of the QL method (TQL2). The loss of accuracy can be attributed to TINVIT's choice of starting vectors and to its iteration stopping criterion.

In this paper, we introduce a new implementation of TINVIT that computes each eigenvector from a different random starting vector and performs an additional iteration after the stopping criterion is satisfied. We present a statistical analysis and the results of numerical experiments with matrices of order up to 525 to show that the numerical accuracy of this new implementation is competitive with that of the implementations of the divide and conquer and QL methods.

Improving the Accuracy of Inverse Iteration

Elizabeth R. Jessup
Ilse C.F. Ipsen

Research Report YALEU/DCS/RR-767
March 1990

The work presented in this paper was supported by DARPA under contract N00014-88-K-0573.
Approved for public release: distribution is unlimited.

1 Introduction

It is our goal to determine an accurate method for computing all eigenvalues and eigenvectors of real symmetric tridiagonal matrices that is efficient both sequentially and in parallel. Experimental results in [13, 14] indicate that bisection with inverse iteration is generally the fastest and most efficient parallel eigensolver on a distributed-memory hypercube multiprocessor such as the INTEL iPSC and that it is also the fastest sequential method for many problems. The computed eigendecompositions, however, are less accurate than those computed by existing implementations of Cuppen's divide and conquer method (TREEQL) [4, 10] or the QL method (TQL2) [2, 19]. The tested implementations of bisection are based on the EISPACK routine BISECT [19] which produces eigenvalues to high *absolute* accuracy [19] and which with minor modification would produce eigenvalues to high *relative* accuracy [6]. The loss of accuracy can thus be attributed to the tested implementation of inverse iteration (EISPACK's TINVIT). In this paper, we identify the factors influencing the accuracy of inverse iteration and present a new implementation of inverse iteration that computes eigenvectors to high absolute accuracy.

Suppose that T is an $n \times n$ real symmetric tridiagonal matrix with eigendecomposition

$$T = U\Lambda U^T, \qquad \Lambda = \begin{pmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{pmatrix}, \quad U = (\, u_1 \quad \dots \quad u_n \,),$$

where the diagonal elements Λ are the eigenvalues of T and the column u_i of the orthogonal matrix U is the eigenvector associated with eigenvalue λ_i. Given an accurately computed eigenvalue $\hat{\lambda}_j$, inverse iteration computes the corresponding eigenvector u_j by performing the power method with the shifted matrix $(T - \hat{\lambda}_j I)^{-1}$:

Algorithm 1.1 (Inverse Iteration)

```
Select a starting vector $y^{(0)}$.
For $k = 1, 2, \ldots$ until convergence:
    Solve $(T - \hat{\lambda}_j I)y^{(k)} = y^{(k-1)}$ for $y^{(k)}$.
$\hat{u}_j = y^{(k)}/\| y^{(k)} \|_2$
```

Representing the starting vector $y^{(0)}$ as a linear combination of the eigenvectors $y^{(0)} = \sum_{i=1}^{n} \eta_i u_i$ gives for the first iterate $y^{(1)} = \sum_{i=1}^{n} \frac{\eta_i}{\lambda_i - \hat{\lambda}_j} u_i$. If the contribution η_j of u_j in $y^{(0)}$ is not too small and if $\hat{\lambda}_j$ is close to λ_j, the contribution $\eta_j/(\lambda_j - \hat{\lambda}_j)$ of u_j in the next iterate $y^{(1)}$ is large, and $y^{(1)}$ is a better approximation to u_j than is $y^{(0)}$ [21], p.321. Likewise, in the next iteration, the contribution of u_j in $y^{(2)}$ increases to $\eta_j/(\lambda_j - \hat{\lambda}_j)^2$ and so on for subsequent iterations. Thus, the iterates $y^{(k)}$ usually converge to u_j in only a few iterations.

If all eigenvalues are well-separated and if the starting vector in each eigenvector computation contains a large enough component η_i, inverse iteration using shifts $\hat{\lambda}_1, \ldots, \hat{\lambda}_n$ in turn computes an orthogonal set of eigenvectors. However, if some eigenvalues are close together, inverse iteration as outlined in Algorithm 1.1 produces eigenvectors that are not orthogonal. An additional step to orthogonalize iterates associated with close eigenvalues is then necessary.

This discussion of the inverse iteration algorithm shows that if the eigenvalues $\hat{\lambda}_i$ are determined to working precision (which is true for BISECT) and if the linear system solution and the orthogonalization of iterates corresponding to close eigenvalues are carried out accurately (which is true for TINVIT), the overall accuracy of inverse iteration is determined by:

1. the choice of starting vector,
2. the reorthogonalization criterion, and
3. the iteration stopping criterion.

We will examine the EISPACK routine TINVIT with regard to each factor in turn and gradually improve its accuracy to that of the implementations of the QL and Cuppen's divide and conquer methods. The numerical experiments involve matrix orders up to $n = 525$, and the conclusions drawn from them therefore may not apply to much larger matrix orders.

This paper is organized as follows. A simple perturbation result is presented in Section 2 to define measures of high absolute accuracy for the computed eigenvalue decomposition. The EISPACK implementation TINVIT is described in Section 3, and its lack of numerical accuracy explored in Section 4. A new implementation of inverse iteration based on the experiments in Section 4 is presented in Section 5. The numerical accuracy of this improved implementation is compared to implementations of Cuppen's divide and conquer method and of the QL method in Section 6. The use of random starting vectors in the new implementation of inverse iteration is justified by a statistical analysis in Section 7.

2 The Computed Eigendecomposition

This section shows that the computed eigendecomposition has high absolute accuracy if its residual and the deviation of the eigenvectors from orthogonality are small.

Suppose that the diagonal elements of Λ are the eigenvalues of $T = U\Lambda U^T$ in descending order

$$\lambda_1 \geq \ldots \geq \lambda_n,$$

and that the column u_i of the orthogonal matrix U is the eigenvector associated with λ_i satisfying $\| u_i \|_2 = 1$. The spectral radius of T is denoted by

$$|\lambda|_{max} \equiv \max\{|\lambda_1|, |\lambda_n|\}.$$

It is further assumed throughout the paper that the matrix T is unreduced, that is, that none of the immediate sub- or superdiagonal elements of T is zero. Otherwise, the matrix would consist of a direct product of disjoint, lower order matrices whose eigendecompositions can be computed independently [21], p.315. Although an unreduced tridiagonal matrix has distinct eigenvalues in exact arithmetic, it may still have computationally coincident ones in finite precision.

Assume that the computed eigenvalues satisfy the same order as the corresponding exact eigenvalues, *i.e.*,

$$\hat{\lambda}_1 \geq \ldots \geq \hat{\lambda}_n.$$

The accuracy of the computed eigendecomposition $\hat{U}\hat{\Lambda}\hat{U}^T$ of T is then determined by the largest residual error $\mathcal{R}$ of any computed eigenpair and by the deviation from orthogonality $\mathcal{O}$ of the computed eigenvectors:

$$\mathcal{R} = \frac{1}{|\hat{\lambda}|_{max}} \max_{1 \leq i \leq n} \| T\hat{u}_i - \hat{\lambda}_i \hat{u}_i \|_2, \qquad \mathcal{O} = \| \hat{U}^T\hat{U} - I \|_\infty.$$

The particular norms for $\mathcal{R}$ and $\mathcal{O}$ were chosen because they are convenient to analyze and to compute. The outcome of the numerical experiments in the later sections does not change if the matrix norm $\mathcal{O}$ is replaced by the vector norm $\max_{1 \leq i \leq n} \| (\hat{U}^T\hat{U} - I)e_i \|_\infty$. Our analysis is restricted to the above norm-based criteria; other quality measures that are applicable when T is known to very high accuracy are discussed in [1, 5, 6, 7].

Lemma 2.1 *If V is a square matrix and*

$$E = V^T V - I, \qquad \bar{E} = VV^T - I$$

then $\| E \|_2 = \| \bar{E} \|_2$.

Proof: Let $V = Y\Sigma X^T$ be the singular value decomposition of V, then

$$V^T V - I = X\Sigma^2 X^T - I = X\Sigma^2 X^T - XX^T = X(\Sigma^2 - I)X^T.$$

Similarly, $VV^T - I = Y(\Sigma^2 - I)Y^T$. Because X and Y are orthogonal matrices,

$$\| E \|_2 = \| V^T V - I \|_2 = \| VV^T - I \|_2 = \| \bar{E} \|_2.$$

■

Theorem 2.1 below shows that the computed eigendecomposition $\hat{U}\hat{\Lambda}\hat{U}^T$ is the exact eigendecomposition of a matrix $T + E$ close to T if residuals and deviation from orthogonality are small. The error matrix E is in general neither symmetric nor tridiagonal.

Theorem 2.1 *Let $\hat{U}\hat{\Lambda}\hat{U}^T$ be the computed eigendecomposition of a symmetric tridiagonal matrix T and let*

$$\mathcal{R} = \frac{1}{|\hat{\lambda}|_{max}} \max_{1 \le i \le n} \| T\hat{u}_i - \hat{\lambda}_i \hat{u}_i \|_2, \qquad \mathcal{O} = \| \hat{U}^T \hat{U} - I \|_\infty.$$

If $\mathcal{R} \le \epsilon_1$ and $\mathcal{O} \le \epsilon_2$, then there exists a matrix E such that

$$T + E = \hat{U}\hat{\Lambda}\hat{U}^T, \qquad \| E \|_2 \le \sqrt{n}\left(|\hat{\lambda}|_{max}\epsilon_2 + |\hat{\lambda}|_{max}\epsilon_1 \sqrt{1 + \sqrt{n}\epsilon_2} \right),$$

where $|\hat{\lambda}|_{max} = \max\{|\hat{\lambda}_1|, |\hat{\lambda}_n|\}$.

Proof: Let

$$E_1 = \frac{1}{|\hat{\lambda}|_{max}}(T\hat{U} - \hat{U}\hat{\Lambda}), \qquad E_2 = \hat{U}^T\hat{U} - I, \qquad \bar{E}_2 = \hat{U}\hat{U}^T - I.$$

Because for any $n \times n$ matrix A, $\| A \|_2 \leq \sqrt{n} \max_{1 \leq i \leq n} \| Ae_i \|_2$ [11], where e_i is the ith canonical vector,

$$\begin{aligned} \| E_1 \|_2 \leq \sqrt{n} \max_{1 \leq i \leq n} \| E_1 e_i \|_2 = \sqrt{n}\mathcal{R} \leq \sqrt{n}\epsilon_1, \\ \| E_2 \|_2 \leq \sqrt{n} \| E_2 \|_\infty = \sqrt{n}\mathcal{O} \leq \sqrt{n}\epsilon_2. \end{aligned}$$

Lemma 2.1 is now used to bound $\| \hat{U}^T \|_2$:

$$\sqrt{n}\epsilon_2 \geq \| E_2 \|_2 = \| \bar{E}_2 \|_2 = \| \hat{U}\hat{U}^T - I \|_2 \geq \| \hat{U}\hat{U}^T \|_2 - \| I \|_2 = \|\hat{U}^T\|_2^2 - 1,$$

where the last equality follows from $\| A^T A \|_2 = \|A\|_2^2$. Therefore, $\|\hat{U}^T\|_2^2 \leq 1 + \sqrt{n}\epsilon_2$.

From the definition of E_1,

$$T\hat{U}\hat{U}^T - \hat{U}\hat{\Lambda}\hat{U}^T = |\hat{\lambda}|_{max} E_1 \hat{U}^T,$$

so that

$$\hat{U}\hat{\Lambda}\hat{U}^T = T\hat{U}\hat{U}^T - |\hat{\lambda}|_{max} E_1 \hat{U}^T = T(I + \bar{E}_2) - |\hat{\lambda}|_{max} E_1 \hat{U}^T = T + E,$$

where $E = T\bar{E}_2 - |\hat{\lambda}|_{max} E_1 \hat{U}^T$, and

$$\begin{aligned} \| E \|_2 & \leq |\lambda|_{max} \| \bar{E}_2 \|_2 + |\hat{\lambda}|_{max} \| E_1 \|_2 \| \hat{U}^T \|_2 \\ & \leq \sqrt{n} \left(|\lambda|_{max}\epsilon_2 + |\hat{\lambda}|_{max}\epsilon_1 \sqrt{1 + \sqrt{n}\epsilon_2} \right). \end{aligned}$$

■

Under the assumptions

$$\| T\hat{U} - \hat{U}\hat{\Lambda} \|_2 \leq \epsilon_1, \qquad \| \hat{U}^T\hat{U} - I \|_2 \leq \epsilon_2,$$

the error matrix is bounded above by

$$\| E \|_2 \leq |\lambda|_{max}\epsilon_2 + \epsilon_1\sqrt{1 + \epsilon_2},$$

which is independent of the matrix order.

3 The EISPACK Routine TINVIT

The steps for computing all eigenvectors of an unreduced symmetric tridiagonal matrix T by inverse iteration are given in Algorithm 3.1:

Algorithm 3.1 (Basic Implementation of Inverse Iteration)

```
For $j = n, n-1, \ldots, 1$:
    1. Choose a starting vector $y_j$.
    2. Solve $(T - \hat{\lambda}_j I)z_j = y_j$ for $z_j$.
    3. If the reorthogonalization criterion is satisfied,
       orthogonalize $z_j$ against iterates associated with computed eigenvalues
       close to $\hat{\lambda}_j$.
    4. If the stopping criterion is not satisfied,
       set $y_j = z_j$ and go to step 2.
    5. The computed eigenvector is $\hat{u}_j = z_j / \| z_j \|_2$.
```

As suggested in [20], p.143 and [21], p.329, the EISPACK implementation TINVIT performs the linear system solution in step 2 by Gaussian Elimination with partial pivoting and the orthogonalizations in step 3 by the modified Gram-Schmidt algorithm. Because TINVIT yields less accurate eigenvectors than do existing implementations of the QL method (TQL2) [2, 19] or Cuppen's divide and conquer method (TREEQL) [4, 10], the loss in accuracy must be due to one or more of the following three factors: the choice of starting vector, the reorthogonalization criterion, and the stopping criterion. TINVIT deals with these issues as follows.

3.1 The Starting Vector

As argued in Section 1, a good starting vector y_j has a large enough contribution of an eigenvector associated with the current eigenvalue λ_j to yield an iterate with dominant components in the eigenspace associated with λ_j.

Without advance knowledge of the eigenvectors, however, it is difficult to ensure a high quality starting vector. For instance, the canonical basis vectors e_1 and e_n should not be used as starting vectors because they are often nearly orthogonal to some eigenvectors of a symmetric tridiagonal matrix T [20], p.147. The vector of all ones is also a poor choice as it is orthogonal to half of the eigenvectors of a symmetric tridiagonal Toeplitz matrix [12].

Analytic determination of a good starting vector is complicated by the role of roundoff error in inverse iteration. As shown in [18, 20, 21, 22], one or two iterations in finite precision arithmetic are generally sufficient to produce a significant iterate component in the correct direction unless the starting vector is exactly orthogonal to that direction.

In agreement with [20], p.147, TINVIT avoids explicit formation of the starting vector y_j as follows. The tridiagonal system $(T - \hat{\lambda}_j I)z_j = y_j$ is solved by using Gaussian Elimination with partial pivoting to factor the matrix $T - \hat{\lambda}_j I = L_j U_j$. (Throughout this paper, we disregard the permutation matrix for simplicity). The vector y_j is chosen so that the result of the forward substitution equals e, the vector of all ones. That is, the computation $L_j e = y_j$ need never be carried out, and the solution of the first linear system in each eigenvector computation amounts to solving only the second of the two triangular systems $U_j z_j = e$.

When computationally coincident eigenvalues (*i.e.*, eigenvalues that are identical to working precision) are used as shifts, their iterates converge to a single eigenvector and fail to span the whole eigenspace. However, these iterates are very sensitive to the value of $\hat{\lambda}_j$ [21], p.329. Wilkinson suggests that the computationally coincident eigenvalues be slightly perturbed so as to make them distinct and that inverse iteration be used with the perturbed eigenvalues to produce iterates that are linearly independent. The increased distance of $\hat{\lambda}_{i+1}$, ..., $\hat{\lambda}_{i+k}$ from $\hat{\lambda}_i$ should affect only the speed of convergence and not the accuracy of inverse iteration [21], p.329.

TINVIT replaces computationally coincident eigenvalues $\hat{\lambda}_i = \hat{\lambda}_{i-1} = \cdots = \hat{\lambda}_{i-k}$ by

$$\hat{\lambda}_i < \hat{\lambda}_i + \epsilon_M \|T\|_R < \cdots < \hat{\lambda}_i + (k-1)\epsilon_M \|T\|_R,$$

where ϵ_M is machine epsilon, and

$$\| T \|_R \equiv \max_{1 \le j \le n} \{ |\alpha_j| + |\beta_j| \}$$

for a matrix T with diagonal elements $\alpha_1, \ldots, \alpha_n$ and off-diagonal elements $\beta_2, \ldots, \beta_n$, and $\beta_1 \equiv 0$. An unreduced tridiagonal matrix T satisfies $\| T \|_R \le \| T \|_\infty$.

3.2 The Reorthogonalization Criterion

The above strategy for perturbing computationally coincident eigenvalues is intended to produce computed eigenvectors that are linearly independent. To assure orthogonal computed eigenvectors, the iterates associated with close eigenvalues are reorthogonalized against each other. In TINVIT, two adjacent eigenvalues $\hat{\lambda}_j$ and $\hat{\lambda}_{j+1}$ are considered close if

$$\hat{\lambda}_j - \hat{\lambda}_{j+1} < 10^{-3} \| T \|_R.$$

The process of reorthogonalizing the iterates is different in sequential and parallel implementations and proceeds as follows.

In a sequential implementation the eigenvectors are computed successively, according to the ascending order of the eigenvalues. That is, at the time of computation of $\hat{u}_j$, the computation of the eigenvectors $\hat{u}_{j+1}$, ..., $\hat{u}_n$ has already been completed. If a computed eigenvalue $\hat{\lambda}_j$ is close to the computed eigenvalue $\hat{\lambda}_{j+1}$ then the iterate z_j is reorthogonalized against $\hat{u}_{j+1}$ and against all eigenvectors, against which $\hat{u}_{j+1}$ was orthogonalized.

In the parallel implementation of [13], all iterates z_i associated with a set of close eigenvalues are computed simultaneously in lock step. If $\hat{\lambda}_j$ is close to $\hat{\lambda}_{j+1}$ then the iterate z_j is orthogonalized against z_{j+1} and against all iterates, against which z_{j+1} was orthogonalized. Although the sequential and parallel implementations could have been based on identical algorithms, the orthogonalization of z_j against the intermediate iterates z_i is used in the parallel implementation to allow pipelined reorthogonalization of eigenvectors [13].

In the outline of TINVIT below, the data structure CLUSTER(i) contains the indices $i+1, \ldots, i+k$ of all those vectors, against which z_i must be orthogonalized.

3.3 The Stopping Criterion

In [20], p.145, Wilkinson shows that if an iterate z_j has a large norm after reorthogonalization but before normalization, the eigenpair $(\hat{\lambda}_j, z_j)$ has a small residual error. Specifically, the large iterate norm $\epsilon_M \|z_j\|_2 = \Omega(n^{-1/2})$ leads to the small residual $\|(T - \hat{\lambda}_j I)z_j\|_2 = O(\epsilon_M n^{-1/2})$ [20], p.145. Furthermore, the large norm of z_j (after reorthogonalization) indicates that the iterates associated with $\hat{\lambda}_{j+1}, \ldots, \hat{\lambda}_n$ were linearly independent (before reorthogonalization) so that the computed eigenvector $\hat{u}_j = z_j / \| z_j \|_2$ is orthogonal to $\hat{u}_{j+1}, \ldots, \hat{u}_n$. (The connection between large iterate norm and successful orthogonalization by the modified Gram-Schmidt procedure is demonstrated in Section 4.1).

From the perturbation result in Section 2 we can then conclude that $(\hat{\lambda}_j, z_j)$ is an eigenpair of a matrix close to T and hence that z_j is an accurate eigenvector. Because $||z_j||_2 \geq || \; z_j \; ||_\infty$, the two-norm can be replaced by the cheaper infinity norm for convergence testing. Thus, if $|| \; y_j \; ||_\infty = 1$ and $\epsilon_M || \; z_j \; ||_\infty > 1$, then z_j is a good eigenvector approximation. The difficulty lies in determining just how large $|| \; z_j \; ||_\infty$ should be. TINVIT stops iteration if $\epsilon_M || \; z \; ||_\infty \geq 1$ (ignoring scaling factors).

3.4 Implementation of Inverse Iteration

A sketch of the EISPACK routine TINVIT is given as Algorithm 3.2 below. Numerical details such as scaling factors used to prevent overflow are not included. The computed eigenvalues are in descending order $\hat{\lambda}_1 \geq \ldots \geq \hat{\lambda}_n$, and ϵ_M is machine epsilon.

Algorithm 3.2 (Outline of TINVIT)

```
For $j = n, n-1 \ldots, 1$

    0. Perturb computationally coincident eigenvalues:
       if $j < n$ and $\hat{\lambda}_j - \hat{\lambda}_{j+1} \leq 0$, then replace $\hat{\lambda}_j$ with $\hat{\lambda}_{j+1} + \epsilon_M \| T \|_R$.

    1. Initialize the set of eigenvalues close to $\hat{\lambda}_j$: $CLUSTER(j) = \emptyset$.
       If $j < n$ and $\hat{\lambda}_j - \hat{\lambda}_{j+1} < 10^{-3} \| T \|_R$, then
       $CLUSTER(j) = CLUSTER(j+1) \bigcup \{j+1\}$.

    2. Initialize the iterate norm $\sigma_j \equiv 0$.

    3. Loop until $\epsilon_M \sigma_j \geq 1$ (error exit after 5 iterations).

       3.a Factor $(T - \hat{\lambda}_j I) = L_j U_j$.

       3.b If this is the first iteration, solve $U_j z_j = e$,
           otherwise solve $L_j U_j z_j = y_j$.

       3.c Sequential implementation:
           Reorthogonalize $z_j$ against all $\hat{u}_i$ with $i \in CLUSTER(j)$.
           Parallel implementation:
           Reorthogonalize $z_j$ against all $z_i$ with $i \in CLUSTER(j)$.

       3.d Set $\sigma_j = \| z_j \|_\infty$ and $y_j = z_j$.

    4. The computed eigenvector is $\hat{u}_j = y_j / \| y_j \|_2$.
```

4 Experimental Results

The experimental results in [14] show that TQL2 or TREEQL generally yield residuals $\mathcal{R} = \max_i \| T\hat{u}_i - \hat{\lambda}_i \hat{u}_i \|_2$ less than 10^{-14} for matrix orders $n \leq 525$; and deviations from orthogonality $\mathcal{O} = \| \hat{U}^T \hat{U} - I \|_\infty$ less than 10^{-14} for $n \approx 32$, less than 10^{-13} for $n \approx 100$, and less than 10^{-12} for $n \approx 512$ (a similar dependence on the matrix order occurs when the deviation from orthogonality is instead measured by $\mathcal{O} = \max_i \| (\hat{U}^T \hat{U} - I)e_i \|_2$). The EISPACK routine TSTURM (a combination of BISECT and TINVIT) yields respective residuals $\mathcal{R}$ less than 10^{-14}, less than 10^{-13}, and less than 10^{-12} and orthogonality measures $\mathcal{O}$ less than 10^{-12}, 10^{-11}, and 10^{-10} for matrix orders 32, 100, and 512, respectively.

The numerical experiments in this section were designed to determine which features of TINVIT need to be modified so that it is at least as accurate in practice as the QL routine TQL2 [19] and the divide and conquer routine TREEQL [10]. All experiments were performed in double precision on a single Sequent Symmetry S81 processor using the Weitek 1167 floating-point accelerator. The eigenvalues were computed with the EISPACK routine BISECT to working precision. Because we tested only matrices up to orders of $n = 525$, our conclusions may not apply to much larger matrix orders.

This paper presents representative results selected from the ones in [14]. The first test matrix [1,2,1] illustrates the case of matrices without close eigenvalues. The matrix [1,2,1] is a symmetric tridiagonal Toeplitz matrix of order n having twos on the diagonal and ones on the first sub- and superdiagonals. Its exact eigenvalues are well-separated and given by [12]

$$\lambda_j = 2\left(1 + \cos\frac{j\pi}{n+1}\right), \qquad 1 \leq j \leq n.$$

For matrix orders $n \leq 525$, the computed eigenvalues $\hat{\lambda}_i$ are also well-separated.

The second test matrix is the 'glued Wilkinson matrix' W_g^+ and represents one of the most difficult test cases for dealing with groups of close eigenvalues. It is constructed as follows. The 'Wilkinson matrix' W_{21}^+ of order $n = 21$ has diagonal elements $10, 9, \ldots, 1, 0, 1, \ldots, 9, 10$ and immediate off-diagonal elements equal to one. It possesses pairs of eigenvalues that are very close [21], p.309. The spacing between eigenvalues in a pair decreases with increasing magnitude of the eigenvalues, and the eigenvalues in the largest pairs are computationally coincident with regard to double precision. The glued Wilkinson matrix W_g^+ of order $21j$ is formed by placing j copies of W_{21}^+ along the diagonal of the matrix and setting off-diagonal elements equal to 10^{-14} at the positions $\beta_{21}, \beta_{42}, \ldots$ where the submatrices join. For matrix orders $n > 200$, W_g^+ has clusters of eigenvalues near the integers $1, 2, \ldots, \lfloor \frac{n}{2} \rfloor$ [17].

The conclusions drawn from numerical experiments with these two matrix types are supported by tests on random matrices in [14].

4.1 Starting Vectors

In this section, we examine the influence of the starting vector on the accuracy of inverse iteration and on the number of iterations performed. To this end, we use the following vectors as starting vectors for the computation of $\hat{u}_j$:

1. the 'correct' eigenvector $\hat{u}_j$: this starting vector is the eigenvector $\hat{u}_j$ computed by inverse iteration with a random starting vector. Each starting vector $\hat{u}_1, \ldots, \hat{u}_n$ has residuals $\mathcal{R} < 10^{-14}$ for all orders and orthogonalities $\mathcal{O} < 10^{-14}$ for $n \leq 42$, $\mathcal{O} < 10^{-13}$ for $n \leq 105$, and $\mathcal{O} < 10^{-12}$ for $n \leq 525$.

2. $\hat{u}_n + \tau\hat{u}_j$: this linear combination of $\hat{u}_n$ and $\hat{u}_j$ is used as starting vector to compute $\hat{u}_j$ for $1 \le j \le n-1$, and a random starting vector is used to compute $\hat{u}_n$. When $\tau = 0$ and $\hat{\lambda}_{n-1} > \hat{\lambda}_n$, $\hat{u}_n + \tau\hat{u}_j$ is roughly orthogonal to the eigenvectors associated with $\lambda_1, \ldots, \lambda_{n-1}$. Increasing the value of τ amounts to increasing the contribution of the desired eigendirection in the starting vector and thus determines the minimal size of the contribution that is sufficient for convergence.

3. random vectors: these vectors have uniformly distributed pseudorandom components between -1 and 1 generated with the linear congruential random number generator from NETLIB. For each matrix order n, a single $n \times n$ random matrix is generated. In one set of experiments, we use the first column of this matrix as the starting vector for *all* eigenvectors. In the second set of experiments, we use column j of the matrix as the starting vector for the jth eigenvector.

4. the TINVIT starting vector y_j: this starting vector is not computed explicitly. Instead it is assumed to be the right-hand side of the lower triangular system $L_j e = y_j$, where e is the vector of all ones, and $T - \hat{\lambda}_j I = L_j U_j$ is the LU decomposition (disregarding the permutation matrix).

For the purposes of this section, TINVIT was modified to perform the same number of iterations for all eigenvectors. Iteration was continued until the required accuracy was achieved but for not more than five iterations. Computationally coincident eigenvalues were perturbed as in step 0 of Algorithm 3.2 except when different starting vectors were used for each shift. For different random starting vectors, the rate of convergence and accuracy of inverse iteration are preserved even if computationally coincident eigenvalues are not perturbed, that is, even if step 0 of TINVIT is omitted. The following two sections distinguish between the experimental results for the cases of well-separated and close eigenvalues.

4.1.1 Starting Vectors for Matrices with Well-Separated Eigenvalues

Table 1 shows the number of iterations required by inverse iteration to compute the eigenvectors to the same accuracy as TQL2 or TREEQL for each type of starting vector.

High accuracy is achieved in one iteration only when accurately computed eigenvectors $\hat{u}_j$ are the starting vectors. More than two iterations are needed only for larger n and only when the starting vector is orthogonal or nearly orthogonal to the computed eigenvector ($\tau \leq 10^{-16}$). All other starting vectors require two iterations.

Thus, for matrices [1,2,1] of order $n \leq 512$, a starting vector component η_j of magnitude 10^{-8} in the desired direction u_j suffices for rapid convergence, *i.e.*, two iterations. Performing more iterations than listed in Table 1 does not significantly change the accuracy. These results are supported by numerical experiments on random matrices with minimal eigenvalue spacing of 10^{-4} [14].

In summary, when all eigenvalues are well-separated the performance of inverse iteration does not strongly depend on the starting vector: random starting vectors and the TINVIT starting vector provide a large enough component in the desired direction for fast convergence.

4.1.2 Starting Vectors for Matrices with Groups of Close Eigenvalues

Table 2 shows the number of iterations for the glued Wilkinson matrix W_g^+ with $n = 42$, 105, and 525 for the different starting vectors. As for matrix [1,2,1],

starting vector	$n = 32$ number of iterations for $\mathcal{R} < 10^{-14}$ $\mathcal{O} < 10^{-14}$	$n = 100$ number of iterations for $\mathcal{R} < 10^{-14}$ $\mathcal{O} < 10^{-13}$	$n = 512$ number of iterations for $\mathcal{R} < 10^{-14}$ $\mathcal{O} < 10^{-12}$
$\hat{u}_j$	1	1	1
$\hat{u}_n$	2	2	4
$\hat{u}_n + 10^{-16}\hat{u}_j$	2	2	3
$\hat{u}_n + 10^{-8}\hat{u}_j$	2	2	2
$\hat{u}_n + 10^{-2}\hat{u}_j$	2	2	2
same random	2	2	2
different random	2	2	2
TINVIT	2	2	2

Table 1: Number of inverse iterations to compute eigenvector $\hat{u}_j$ for matrix [1,2,1] of order n. The same number of iterations is performed for each $\hat{u}_j$.

accurate eigenvectors are produced in one iteration only when the starting vector is the eigenvector. Two iterations suffice when the starting vector has a correct component of size at least 10^{-8} or when a different random starting vector is used for each eigenvector. The remaining starting vectors require more than two iterations. For $n = 525$, inverse iteration does not converge in five iterations when the iterations are started with $\hat{u}_n$, $\hat{u}_n + 10^{-16}\hat{u}_j$ or with the TINVIT starting vector.

Table 3 illustrates the connection between the convergence rate of the iterates and their linear dependence for different types of starting vectors and the matrix W_g^+. The numbers in Table 3 were obtained as follows. The iterates z_j, $1 \leq j \leq n$, before the reorthogonalization step 3.c in the *first* iteration of TINVIT compose the columns of an $n \times n$ matrix. The smallest singular value of this matrix is listed in the first column of numbers, and the smallest norm σ_j attained by the z_j, $1 \leq j \leq n$, after reorthogonalizing in step 3.c is listed in the second column of numbers. The same information is given for z_j in the second iteration of TINVIT in the last two columns. These data show that except in the case of different random starting vectors, the iterates after the first iteration are linearly dependent. Thus, the modified Gram-Schmidt procedure breaks down and produces vectors that are almost zero. Likewise, the second iteration fails to produce linearly independent iterates for all but the different random starting vectors. (The singular value for the matrix of iterates from the same random starting vector is so small, 10^{-18}, that the iterates can be considered numerically linearly dependent).

While the TINVIT starting vectors are difficult to analyze, the other choices suggest a possible correlation between linearly dependent starting vectors and iterates: linearly dependent starting vectors lead to linearly dependent iterates

starting vector	$n = 42$ number of iterations for $\mathcal{R} < 10^{-14}$ $\mathcal{O} < 10^{-14}$	$n = 105$ number of iterations for $\mathcal{R} < 10^{-14}$ $\mathcal{O} < 10^{-13}$	$n = 525$ number of iterations for $\mathcal{R} < 10^{-14}$ $\mathcal{O} < 10^{-12}$
$\hat{u}_j$	1	1	1
$\hat{u}_n$	3	3	> 5
$\hat{u}_n + 10^{-16}\hat{u}_j$	3	3	> 5
$\hat{u}_n + 10^{-8}\hat{u}_j$	2	2	2
$\hat{u}_n + 10^{-2}\hat{u}_j$	2	2	2
same random	2	3	3
different random	2	2	2
TINVIT	2	3	> 5

Table 2: Number of inverse iterations to compute eigenvector $\hat{u}_j$ for the glued Wilkinson matrix W_g^+ of order n. The same number of iterations is performed for each $\hat{u}_j$.

starting vector	smallest singular value of first iterates	minimal iterate norm $\min_j \| z_j \|_\infty$ after one iteration	smallest singular value of second iterates	minimal iterate norm $\min_j \| z_j \|_\infty$ after two iterations
$\hat{u}_n$	0	0	0	$1.24d-13$
same random vector	0	$4.69d-12$	10^{-18}	$7.04d-04$
different random vector	0.02	$4.94d-04$	0.08	> 1.00
TINVIT	0	$4.94d-12$	0	$1.06d-12$

Table 3: Singular values of the matrix of iterates and smallest iterate norm for the glued Wilkinson matrix W_g^+ of order $n = 525$ after one and after two iterations of TINVIT.

matrix order	smallest singular value
42	.0362
100	.0341
105	.0198
512	1.d-229
525	.0128

Table 4: The smallest singular value for a matrix of different random starting vectors.

in the case of computationally coincident eigenvalues. Table 4 shows that the smallest singular value for a matrix composed of n different random starting vectors is much larger than zero; the only exception is $n = 512$ where the 512th column is linearly dependent on the first 479. Because none of the test matrices has a group of eigenvalues including both the 479th and the 512th eigenvalues, inverse iteration starts out with linearly independent random starting vectors for all iterates associated with the same group of close eigenvalues.

In summary, when a different random starting vector is used to compute each eigenvector of the glued Wilkinson matrix, both the starting vectors and the iterates are highly likely to be linearly independent. This correlation between linear dependence of starting vectors and number of iterations can be observed to a lesser degree for other large matrices with groups of close eigenvalues [14].

4.2 Stopping Criterion

The experimental results in this section show that TINVIT's choice of stopping criterion causes inverse iteration to stop before highest accuracy is attained. We will examine an alternative that consistently improves the accuracy. For the experiments in this section, TINVIT was modified to compute each eigenvector from a different random starting vector and to use unperturbed computed eigenvalues as shifts.

matrix	iteration	minimal iterate norm $\min_j \| z_j \|_\infty$	maximal residual $\mathcal{R}$	deviation from orthogonality $\mathcal{O}$
$[1, 2, 1]$ $n = 100$	1	> 1.00	$3.18d - 14$	$3.05d - 12$
	2	> 1.00	$1.58d - 16$	$3.20d - 14$
W_g^+ $n = 105$	1	0.14	$1.47d - 13$	$1.68d - 11$
	2	> 1.00	$8.70d - 16$	$3.85d - 15$
$[1, 2, 1]$ $n = 512$	1	> 1.00	$1.60d - 11$	$4.62d - 09$
	2	> 1.00	$3.93d - 16$	$1.57d - 13$
W_g^+ $n = 525$	1	$4.94d - 04$	$3.42d - 09$	$6.02d - 07$
	2	> 1.00	$5.99d - 15$	$1.98d - 14$

Table 5: Iterate norm, residual, and orthogonality for matrices [1,2,1] and W_g^+ after one and after two inverse iterations. A different random starting vector is used for each eigenvector computation.

For matrices [1,2,1] and W_g^+, Table 5 shows how the accuracy of the computed eigendecomposition depends on the norm of the computed iterates (after reorthogonalization in step 3.c of Algorithm 3.2). The TINVIT stopping criterion works correctly for both orders of the glued Wilkinson matrix W_g^+: unit iterate norm and full accuracy are both attained on the second iteration. It fails, however, on the matrix [1,2,1] where all iterates have greater than unit norm but less than full accuracy on the first iteration. The same conclusions can be drawn from experiments with random matrices in [14]. It seems, therefore, that at least two iterations should always be performed regardless of iterate norm when different random starting vectors are used. In other words, after the iterate norm is large enough and the loop in step 3 of TINVIT is exited, perform one more iteration. The additional iteration was already suggested in [21], p.324, but was not implemented in TINVIT.

We have not found a simple correlation between size of the iterate norms and the number or size of the groups of close eigenvalues.

4.3 Reorthogonalization

For the purposes of this section, TINVIT was modified to compute each eigenvector from a different random starting vector and to perform one more iteration after the iterate norm becomes large enough, *i.e.*, one more iteration after exiting the loop in step 3 of TINVIT. In the numerical experiments below, we vary the distance at which adjacent eigenvalues are considered to be so close as to require orthogonalization of the associated iterates.

Table 6 shows the residuals $\mathcal{R}$ and deviations from orthogonality $\mathcal{O}$ for matrix [1,2,1] of order $n = 100$ as the reorthogonalization criterion is varied from 0 to $10^{10} \| T \|_R$ after one and after two inverse iterations. These data confirm that more orthogonalization is not a substitute for extra iterations because

criterion	number of vectors orthog-onalized	one iteration		two iterations	
		$\mathcal{R}$	$\mathcal{O}$	$\mathcal{R}$	$\mathcal{O}$
$10^{10}\| T \|_R$ (all)	99	6.09d-13	6.41d-15	2.12d-16	5.75d-15
$10^{-1}\| T \|_R$	97	1.87d-12	7.40d-15	2.13d-16	6.12d-15
$10^{-2}\| T \|_R$	33	7.69d-14	4.97d-12	1.67d-16	1.82d-15
$10^{-3}\| T \|_R$	2	3.18d-13	3.05d-12	1.58d-16	3.20d-14
$10^{-5}\| T \|_R$	1	1.10d-13	2.75d-11	1.90d-16	4.28d-14
0	0	2.28d-13	2.68d-11	1.61d-16	4.68d-14

Table 6: Accuracy for matrix $[1, 2, 1]$ with different reorthogonalization criteria when $n = 100$.

small residuals are not attained until the second iteration even with reorthogonalization of all eigenvectors. Table 7 shows the same situation for W_g^+ when $n = 105$ and $n = 525$, as well as the fraction of inverse iteration time spent in the modified Gram-Schmidt procedure.

Increasing the reorthogonalization criterion beyond that of TINVIT does not significantly improve the accuracy for matrices $[1, 2, 1]$ and W_g^+. It can, however, substantially increase the computation time. With the TINVIT criterion $10^{-3}\|T\|_R$, most of the eigenvectors of W_g^+ are reorthogonalized (80% when $n = 105$ and 96% when $n = 525$), but reorthogonalization occurs in many small groups. In contrast, with the criterion $10^{10}\|T\|_R$ all eigenvectors are reorthogonalized as one group, and the cost rises markedly although the accuracy hardly changes.

These experiments show that the best possible orthogonality can generally

order	criterion	$\mathcal{R}$	$\mathcal{O}$	number of vectors	time for MGS	fraction MGS time
$n = 105$	$10^{10}\| T \|_R$ (all)	1.97d-16	4.57d-15	104	30.6	.67
	$10^{-1}\| T \|_R$	1.60d-16	3.14d-15	88	27.7	.10
	$10^{-3}\| T \|_R$	8.70d-16	3.85d-15	84	1.4	.07
	$10^{-5}\| T \|_R$	2.09d-16	2.52d-13	78	1.4	.05
	0	1.85d-16	2.05	0	0	0
$n = 525$	$10^{10}\| T \|_R$ (all)	7.58d-15	1.53d-14	524	5807.1	.98
	$10^{-1}\| T \|_R$	4.69d-15	3.51d-14	523	5287.6	.73
	$10^{-3}\| T \|_R$	5.99d-15	1.98d-14	504	338.1	.15
	$10^{-5}\| T \|_R$	3.46d-15	6.24d-13	498	253.1	.12
	0	2.04d-16	6.38	0	0	0

Table 7: Accuracy and computation time for W_g^+ after two inverse iterations with different reorthogonalization criteria when $n = 100$ and $n = 525$. The last column shows the fraction of reorthogonalization (MGS) time in inverse iteration.

be attained only by the time-consuming process of reorthogonalizing all eigenvectors. The accuracy desired here, however, can usually be achieved by means of the TINVIT reorthogonalization criterion along with different random starting vectors and the improved stopping criterion of Section 4.3.

5 A New Implementation of Inverse Iteration

The improvements to inverse iteration developed in Section 4 are incorporated into the following algorithm. These changes are based on experiments in Section 4 with matrix orders $n \leq 525$ and may not apply to much larger matrix orders.

Algorithm 5.1 (Improved Inverse Iteration Algorithm (III))

```
For $j = n, n-1, \ldots, 1$
    1. Initialize the set of eigenvalues close to $\hat{\lambda}_j$: $CLUSTER(j) = \emptyset$.
       If $j < n$ and $\hat{\lambda}_j - \hat{\lambda}_{j+1} < 10^{-3} \| T \|_R$, then
       $CLUSTER(j) = CLUSTER(j+1) \bigcup \{j+1\}$.
    2. Generate a random vector $x_j$ with uniformly distributed components
       in the interval [-1,1], and form the starting vector $y_j = x_j / \| x_j \|_2$.
    3. Initialize the iterate norm $\sigma_j \equiv 0$.
    4. Loop until $\epsilon_M \sigma_j \geq 1$ (error exit after 5 iterations).
        4.a Solve $(T - \hat{\lambda}_j I) z_j = y_j$.
        4.b Sequential implementation:
            Reorthogonalize $z_j$ against all $\hat{u}_i$ with $i \in CLUSTER(j)$.
            Parallel implementation:
            Reorthogonalize $z_j$ against all $z_i$ with $i \in CLUSTER(j)$.
        4.c Set $\sigma_j = \| z_j \|_\infty$ and $y_j = z_j$.
    5. Repeat step 4 once.
    6. The computed eigenvector is $\hat{u}_j = y_j / \| y_j \|_2$.
```

Tables 8 and 9 compare the computation time of the EISPACK routine TSTURM with that of the EISPACK routine BISECT combined with algorithm III (B/III). Because of the additional iterations performed, the computation time of III is substantially higher than that of TINVIT. For matrix [1,2,1] of

n		TSTURM			B/III		
	time to compute eigenvalues (seconds)	time to compute eigenvectors (seconds)	$\mathcal{R}$	$\mathcal{O}$	time to compute eigenvectors (seconds)	$\mathcal{R}$	$\mathcal{O}$
32	1.1	0.3	4.15d-15	4.00d-13	0.4	1.30d-16	4.27d-15
100	11.3	2.0	2.46d-14	8.48d-12	3.2	1.56d-16	3.15d-14
512	276.7	72.2	1.26d-13	4.48d-11	125.8	4.11d-16	1.78d-13

Table 8: Times, residuals, and orthogonalities for eigensystems computed by TSTURM and by B/III for matrix $[1, 2, 1]$.

order $n = 512$, however, very little orthogonalization of eigenvectors takes place (see Table 6), and eigenvector computation is cheap compared to eigenvalue computation. The longer time of algorithm III represents only a 13% increase in total computation time for B/III over TSTURM.

The storage requirements for algorithm III are the same as for TINVIT. The time for generation of random starting vectors in algorithm III is small compared to the total computation time. It constitutes less than 4% of the total eigenvector computation time for matrix [1,2,1] of order $n \leq 512$ and for W_g^+ of order $n \leq 525$. A 1000 x 1000 matrix of random elements can be generated in 14.00 seconds.

6 Comparison with Other Methods

This section offers an experimental comparison of Cuppen's divide and conquer method, the QL method, and bisection with inverse iteration. The respective

n		TSTURM			B/III		
	time to compute eigenvalues (seconds)	time to compute eigenvectors (seconds)	$\mathcal{R}$	$\mathcal{O}$	time to compute eigenvectors (seconds)	$\mathcal{R}$	$\mathcal{O}$
42	1.6	0.4	4.25d-15	2.3d-13	0.6	1.61d-16	2.61d-15
105	4.72	3.0	5.11d-14	2.36d-12	4.8	6.98d-16	4.43d-15
525	23.3	171.1	1.14d-13	4.08d-11	333.4	5.55d-15	1.69d-14

Table 9: Times, residuals, and orthogonalities for eigensystems computed by TSTURM and by B/III for matrix W_g^+.

implementations are TREEQL [10], TQL2 [19], and B/III. TREEQL switches from divide and conquer to TQL2 for subproblems of order $n \leq 50$. For a given problem, the relative speeds of the three methods depend on the degree of deflation in TREEQL, the amount of matrix splitting in TQL2, and the clustering of eigenvalues for B/III. Because they illustrate the range of results for all matrices from [14], we use the three test matrices [1,2,1], W_g^+, and [1,u,1] as a basis for comparison. The matrix [1,u,1] has ones in its first subdiagonal and superdiagonal and the value $i \times 10^{-6}$ in the ith diagonal position. It undergoes little deflation when its eigendecomposition is computed by TREEQL. As none of these test matrices contains row sums of widely differing magnitudes, we exclude IMTQL2 from the comparison: the performance and accuracy of TQL2 and IMTQL2 are nearly identical for these test matrices [14].

Table 10 shows that the maximal residual $\mathcal{R} = \max_{1 \leq i \leq n} \| T\hat{u}_i - \hat{\lambda}_i \hat{u}_i \|_2$ and deviation from orthogonality $\mathcal{O} = \| \hat{U}^T \hat{U} - I \|_\infty$ of the eigendecompositions computed by the three methods do not differ significantly.

matrix order	method	maximal residual $\mathcal{R}$	deviation from orthogonality $\mathcal{O}$
n = 32 or 42	TREEQL	3.26d-15	5.59d-15
	TQL2	1.52d-15	1.30d-14
	B/III	1.80d-16	6.20d-15
n = 100 or 105	TREEQL	6.07d-14	2.75d-15
	TQL2	2.39d-15	1.06d-14
	B/III	3.67d-15	8.52d-14
n = 512 or 525	TREEQL	3.96d-15	1.67d-13
	TQL2	1.66d-14	2.50d-13
	B/III	6.05d-15	7.92d-13

Table 10: Maximal residual and orthogonality of eigendecompositions computed by B/III, TREEQL, and TQL2 for matrices [1,2,1], W_g^+, [1,u,1].

As shown in Figures 1–3, however, the computation times for the problems can differ significantly. The top graphs in these figures show the different computation times for matrix orders $n \leq 60$. B/III is slowest for matrices [1,2,1] and [1,u,1] of order $n \leq 20$. TQL2 is fastest for [1,2,1] and [1,u,1] of order $n \leq 20$ and slowest for all matrices of order $50 \leq n \leq 525$. TREEQL is fastest for W_g^+ of all orders and for [1,2,1] of order $20 \leq n \leq 60$ due to moderate ([1,2,1]) and heavy (W_g^+) deflation. The bottom graphs in Figures 1–3 show the different computation times for matrix orders $60 \leq n \leq 512$. For $n = 512$, TREEQL is about 2 to 40 times faster than TQL2, and B/III is about eight times faster than TQL2.

Because the degree of deflation and the grouping of eigenvalues are rarely known in advance, it is generally not possible to select the fastest serial method for a given problem. In all of our experiments, however, B/III is much faster than TQL2 and equally accurate. For larger matrix orders, B/III is fastest for light to medium deflation, while TREEQL is fastest for heavy deflation.

7 A Statistical Analysis of Inverse Iteration

The preceding sections experimentally establish the design choices for an accurate implementation of inverse iteration. Because they rely on the use of starting vectors with randomly distributed components, we now give a statistical analysis to explain some of the experimental observations. We proceed as follows. Section 7.1 states our assumptions; Section 7.2 defines a good eigenvector approximation; Section 7.3 determines the expected quality of a random starting vector and briefly discusses the limitations of the analysis, and Section 7.4 estimates the error in applying the analysis based on starting vectors with normally distributed components to starting vectors with uniformly distributed

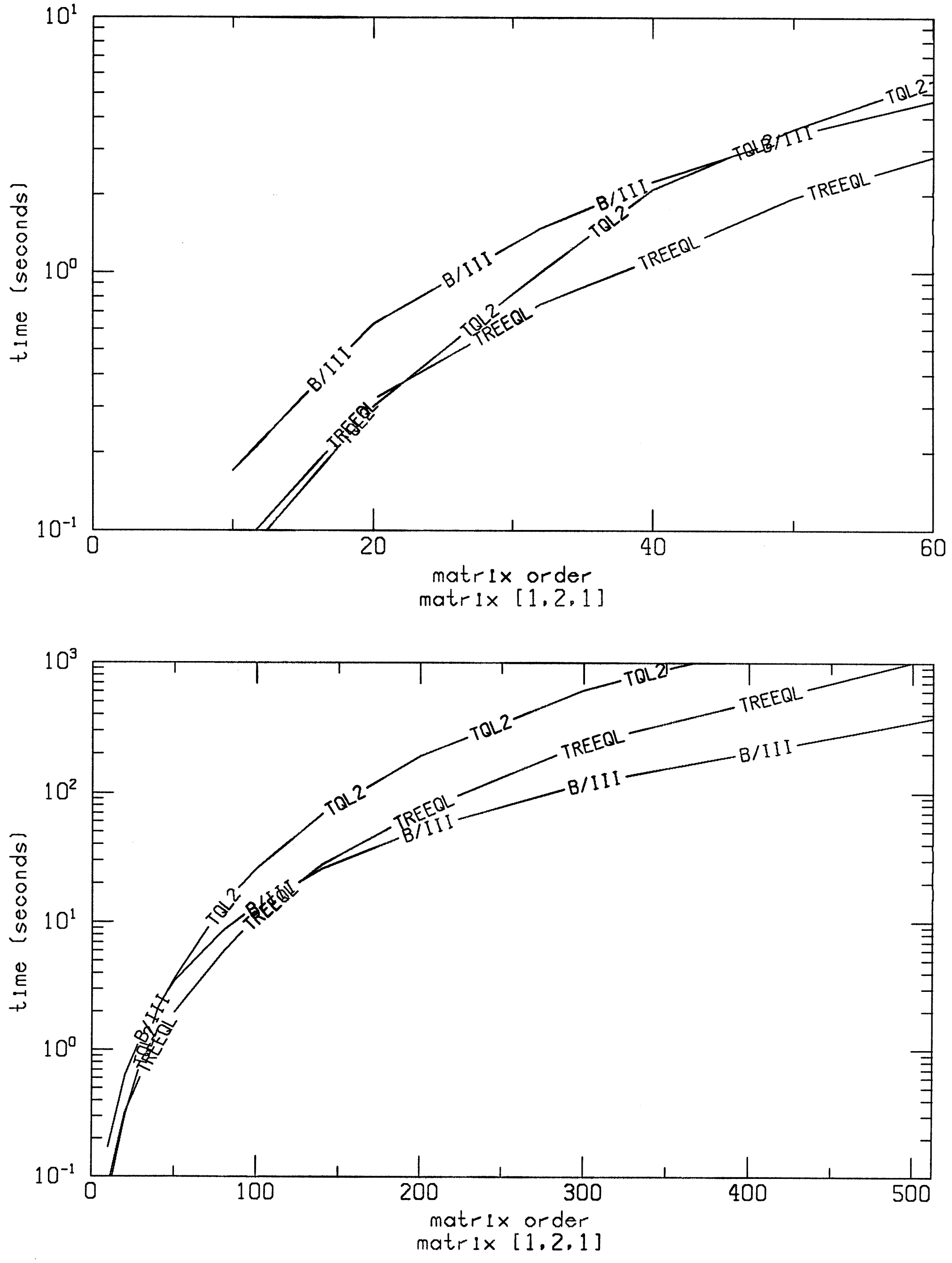

Figure 1: Times for TQL2, TREEQL, and B/III *versus* matrix order for matrix [1,2,1].

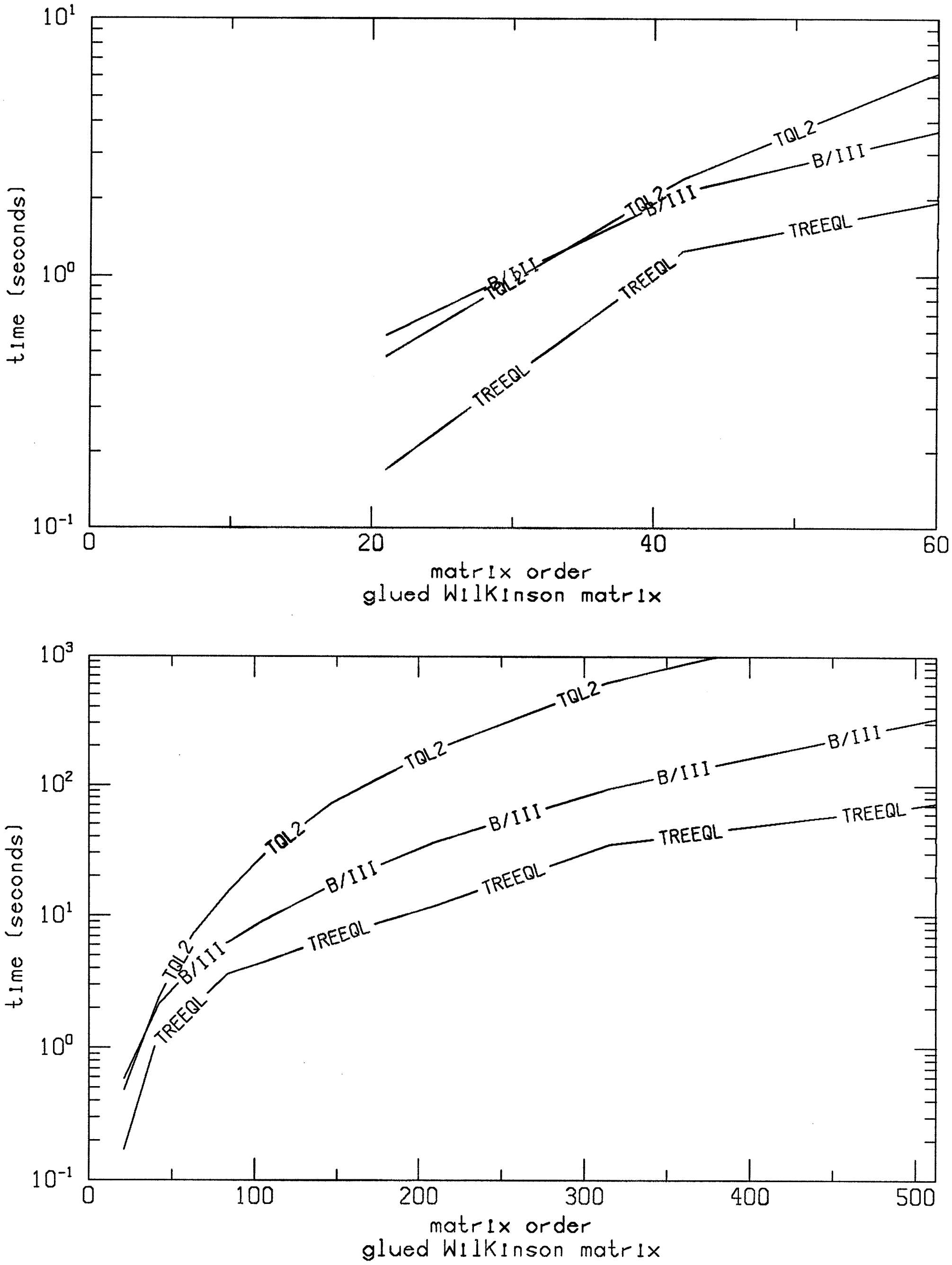

Figure 2: Times for TQL2, TREEQL, and B/III *versus* matrix order for the glued Wilkinson matrix W_g^+.

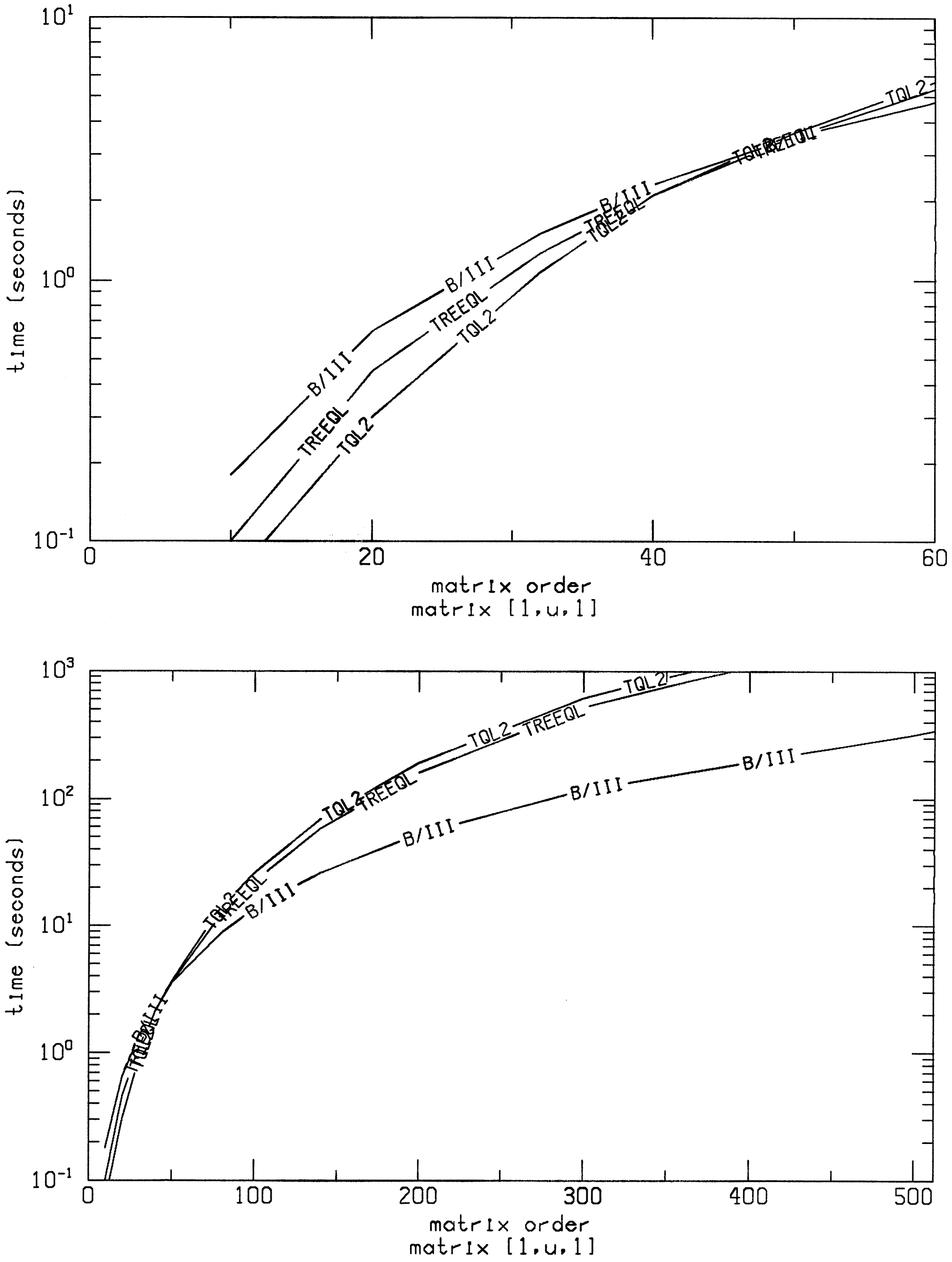

Figure 3: Times for TQL2, TREEQL, and B/III *versus* matrix order for matrix [1,u,1].

components. Statistical analyses of methods for computing eigen*values* can be found, for example, in [9, 16].

7.1 Assumptions

A unit-norm starting vector $y = x/\| x \|_2$ is computed from a vector x with independent random components, each of which has a normal distribution with mean 0 and variance 1 (normal (0,1)). Such vectors y are uniformly distributed on the unit n-sphere [8]. Because the distribution of their components is invariant under orthogonal transformations, these vectors can be represented in terms of the orthonormal basis of eigenvectors $\{u_1, u_2, \ldots, u_n\}$ of the symmetric tridiagonal matrix T:

$$y = (\eta_1, \eta_2, \ldots, \eta_n)^T, \qquad \text{where } \; y = \sum_{i=1}^{n} \eta_i u_i, \quad \| u_i \|_2 = 1, \quad \sum_{i=1}^{n} \eta_i^2 = 1.$$

7.2 The Quality of an Approximate Eigenvector

A vector y as defined above is a good approximation to u_i if η_i is much larger than any other component, *i.e.* if $\eta_i^2 \geq 1 - \epsilon^2$ for some error tolerance $0 \leq \epsilon \leq 1$. Geometrically, the angle θ_i between y and u_i satisfies $\cos\theta_i \geq \sqrt{1-\epsilon^2}$. Similarly, y is a good approximation to a linear combination of eigenvectors u_1, ..., u_d if $\sum_{i=1}^{d} \eta_i^2 \geq 1 - \epsilon^2$. Because random vectors are uniformly distributed on the sphere, the probability that y is a good approximation to the linear combination is just the fraction of the surface area of the sphere defined by all vectors whose components $\xi_1, \ldots, \xi_d$ satisfy $\sum_{i=1}^{d} \xi_i^2 \geq 1 - \epsilon^2$.

The probability that $\sum_{i=1}^{d} \eta_i^2 \geq 1 - \epsilon^2$ is determined by integrating the probability density function of the sum $\sum_{i=1}^{d} \eta_i^2$ between $1 - \epsilon^2$ and 1. If the component ξ_i has a normal (0,1) distribution, then $\eta_i^2 = \xi_i^2 / \sum_{j=1}^{n} \xi_j^2$ has a $B(\frac{1}{2}, \frac{n-1}{2})$ distribution, and the sum $\sum_{i=1}^{d} \eta_i^2$ has a $B(\frac{d}{2}, \frac{n-d}{2})$ distribution [8] with probability density function $t^{-\frac{1}{2}}(1-t)^{\frac{n-3}{2}}$. The probability that y is a good approximation to a linear combination of eigenvectors $u_1, \ldots, u_d$ is thus given in the following theorem:

Theorem 7.1 *Let x have independent random components, each with a normal (0,1) distribution, and let $y = x/\| x \|_2 = (\eta_1, \ldots, \eta_n)^T$. Given $0 \le \epsilon \le 1$, the probability that $\sum_{i=1}^{d} \eta_i^2 \ge 1 - \epsilon^2$ is*

$$P(\sum_{i=1}^{d} \eta_i^2 \ge 1 - \epsilon^2) \ge 1 - \alpha \int_0^{1-\epsilon^2} t^{\frac{d}{2}-1}(1-t)^{\frac{n-d-2}{2}} dt \equiv 1 - \alpha \mathcal{I} \qquad (1)$$

with $\alpha = \frac{\gamma(\frac{n}{2})}{\gamma(\frac{d}{2})\gamma(\frac{n-d}{2})}$.

7.3 The Quality of the Starting Vectors

The experiments in Section 4 show that random vectors make good starting vectors for inverse iteration. It turns out that the random starting vectors used were linearly independent and not orthogonal to the eigenvectors being computed. In this section, we demonstrate the practical usefulness of Theorem 7.1 and establish the number of times a random starting vector can be reused for computing eigenvectors associated with well-separated eigenvalues.

For rapid convergence of an iterate to an eigenvector u_i, it is essential that the starting vector y have a large enough component in the u_i direction. The probability that a component η_i is of size at least $\sqrt{1-\epsilon^2}$ is given in Theorem 7.1 with $d = 1$. Table 11 gives these probabilities for matrix orders $n = 100$, 1000, and 10000 for a range of $1 - \epsilon^2$ values. The integral in Theorem 7.1 was computed by Gauss-Legendre quadrature with 100 nodes. For $n \le 10000$, the probability that any one component of y is at least .0001 is no smaller than $1 - 10^{-16}$, while the probability that any one component is at least .001 is no

	for $n = 100$	for $n = 1000$	for $n = 10000$
$\sqrt{1-\epsilon^2}$	$P(\vert\eta_i\vert \geq \sqrt{1-\epsilon^2})$	$P(\vert\eta_i\vert \geq \sqrt{1-\epsilon^2})$	$P(\vert\eta_i\vert \geq \sqrt{1-\epsilon^2})$
$< 10^{-4}$	1.00 (16)	1.00 (16)	1.00 (16)
10^{-3}	0.99	0.97	0.90
10^{-2}	0.92	0.76	0.34
10^{-1}	0.34	0 (2)	0 (16)
$> .7$	0 (16)	0 (16)	0 (16)

Table 11: Lower bounds on the probability that $\eta_i^2 \geq 1 - \epsilon^2$. Numbers in parentheses indicate the number of zero decimal places.

smaller than 0.9. The unit two-norm of the starting vector guarantees that not all components are very small. Recall that a component of size 10^{-8} is sufficient for fast convergence. For a given tolerance $1 - \epsilon^2$, the probability bounds decrease as n increases: as the number of components in a vector increases, the probability that any one component is large decreases.

Table 4 shows that sets of n randomly generated starting n-vectors tend to be linearly independent. We call a set of vectors $\{x_1, \ldots, x_n\}$ *(numerically) linearly dependent up to a tolerance* $\epsilon > 0$ if there exists a set of nonzero coefficients $\alpha_1, \alpha_2, \ldots, \alpha_n$ such that $\sum_{i=1}^n \alpha_i^2 = 1$ and $\| z \|_2 = \| \sum_{i=1}^n \alpha_i x_i \|_2 \leq \epsilon$. The following theorem gives an upper bound on the probability that $x_1, \ldots, x_n$ are linearly dependent up to tolerance ϵ.

Theorem 7.2 (Linear Independence of Starting Vectors) *Assume that the vectors x_i have independent random components with normal (0,1) distributions and that $z = \sum_{i=1}^n \alpha_i x_i$ with $\sum_{i=1}^n \alpha_i^2 = 1$. Given $\epsilon > 0$, the probability that $\| z \|_2 \leq \epsilon$ is bounded above by*

$$P(\| z \|_2 \leq \epsilon) \leq \frac{2n\epsilon}{\sqrt{2\pi}}.$$

Proof: Let $x_i = (\xi_{1i}, \xi_{2i}, \ldots, \xi_{ni})^T$, $1 \leq i \leq n$, where each component ξ_{ij} has a normal (0,1) distribution. If $z = \sum_{i=1}^n \alpha_i x_i = (\zeta_1, \zeta_2, \ldots, \zeta_n)^T$ with $\sum_{i=1}^n \alpha_i^2 =$

1 then the component $\zeta_i = \sum_{i=1}^{n} \alpha_i \xi_{ij}$ has a normal (0,1) distribution and probability density function $f(t) = \frac{1}{\sqrt{2\pi}} e^{-t^2/2}$. This gives an upper bound on $P(\| z \|_\infty \leq \epsilon)$:

$$P(\| z \|_\infty \leq \epsilon) = P(|\zeta_i| \leq \epsilon, 1 \leq i \leq n) \leq \sum_{i=1}^{n} P(|\zeta_i| \leq \epsilon) = nP(|\zeta_1| \leq \epsilon).$$

Because

$$P(|\zeta_1| \leq \epsilon) = P(-\epsilon \leq \zeta_1 \leq \epsilon) \leq P(\zeta_1 \leq \epsilon) - P(\zeta_1 \leq -\epsilon)$$

we have

$$\frac{1}{n} P(\| z \|_\infty \leq \epsilon) \leq \int_{-\infty}^{\epsilon} f(x)dx - \int_{-\infty}^{-\epsilon} f(x)dx = 2\int_{0}^{\epsilon} f(x)dx \leq \frac{2\epsilon}{\sqrt{2\pi}}.$$

Because $\| z \|_\infty \leq \| z \|_2$, $P(\| z \|_\infty \leq \epsilon) \geq P(\| z \|_2 \leq \epsilon)$, and

$$P(\| z \|_2 \leq \epsilon) \leq \frac{2n\epsilon}{\sqrt{2\pi}}.$$

■

For the computation of eigenvectors associated with well-separated eigenvalues, linear independence of the starting vectors seems less important, and one could try to use the same random starting vector for all of them. Given $\epsilon > 0$, the same starting vector y can be used to compute eigenvectors $u_1, \ldots, u_d$ if $\eta_i^2 \geq 1 - \epsilon^2$ for $1 \leq i \leq d$. The following theorem shows how the number d of eigenvectors for which y can be used, depends on the probability ρ with which each of the d eigenvectors provides a significant contribution of y.

Theorem 7.3 (Reuse of Starting Vectors) *Assume that x has independent random components with normal (0,1) distributions and that $y = x/\| x \|_2$. If $d \leq \lfloor \frac{1-\rho}{\alpha \mathcal{I}} \rfloor$, then $\eta_i^2 \geq 1 - \epsilon^2$ for $1 \leq i \leq d$ with probability at least ρ.*

Proof: Let $y = x/\| x \|_2 = (\eta_1, \ldots, \eta_n)^T$. Then

$$\begin{aligned} P(\eta_i^2 \geq 1-\epsilon^2, 1 \leq i \leq d) &= 1 - P(\eta_i^2 < 1-\epsilon^2, 1 \leq i \leq d) \\ &\geq 1 - \sum_{i=1}^{d} P(\eta_i^2 \leq 1-\epsilon^2) = 1 - d\alpha\mathcal{I}, \end{aligned}$$

where the last equality comes from Theorem 7.1. Setting $\rho = 1 - d\alpha\mathcal{I}$ gives $d \leq \lfloor \frac{1-\rho}{\alpha\mathcal{I}} \rfloor$. ■

Table 12 shows values of d for several choices of $\sqrt{1-\epsilon^2}$ when $n = 100$, 1000, 10000: the number of times y can be reused decreases with increasing matrix order n, for fixed ρ and ϵ. This echoes the trend observed in Table 11: a long vector of norm one is less likely to have large components and so is less acceptable for reuse. From the numerical experiments, we know that a component of size 10^{-8} suffices for rapid convergence. According to Table 12 all components are of size 10^{-4} for random vectors up to length 1000 with probability 0.99 so that the same starting vector can be used to compute all eigenvectors for matrices of order up to $n = 10000$ with probability 0.99.

Unfortunately, the applicability of our analysis to the results of inverse iteration is extremely limited. If $z = (T - \hat{\lambda}_j I)^{-k} y = (\zeta_1, \ldots, \zeta_n)^T$ is the unnormalized kth iterate, and no reorthogonalization has taken place, the probability that any one component ζ_j is larger in magnitude than $\sqrt{1-\epsilon^2}$ is [14]

$$P(\zeta_j^2 \geq 1-\epsilon^2) \geq 1 - \alpha \int_0^{(\lambda_j - \hat{\lambda}_j)^{2k}(1-\epsilon^2)} t^{-\frac{1}{2}}(1-t)^{\frac{n-3}{2}}\, dt.$$

According to the mean-value theorem for definite integrals, the integral $\mathcal{I}$ is bounded above by $(\lambda_j - \hat{\lambda}_j)^{2k}(1-\epsilon^2)$. Thus, if $\hat{\lambda}_j$ is a good approximation to λ_j, then $\mathcal{I}$ is small and the probability that z_j approximates u_j is close to one. If $\lambda_j - \hat{\lambda}_j = \lambda_{j+1} - \hat{\lambda}_j$, then z_j approximates u_j and u_{j+1} with equal probability.

ρ	$\sqrt{1-\epsilon^2}$	for $n = 100$	for $n = 1000$	for $n = 10000$
		d	d	d
0.5	$< 10^{-4}$	100	1000	10000
	10^{-3}	50	16	5
	10^{-2}	6	2	< 1
	10^{-1}	< 1	< 1	< 1
0.9	$< 10^{-4}$	100	1000	10000
	10^{-3}	10	3	1
	10^{-2}	1	< 1	< 1
	10^{-1}	< 1	< 1	< 1
0.99	$< 10^{-4}$	100	1000	10000
	10^{-3}	1	< 1	< 1
	10^{-2}	< 1	< 1	< 1
	10^{-1}	< 1	< 1	< 1

Table 12: The number of times d a starting vector can be used, when $\eta_i^2 \geq 1 - \epsilon^2$ for $1 \leq i \leq d$ with probability ρ.

When λ_j and λ_{j+1} are close but not equal, one of the two eigenvectors u_j and u_{j+1} will be approximated better than the other only if $(1-\epsilon^2)(\lambda_j - \hat{\lambda}_j)$ and $(1-\epsilon^2)(\lambda_{j+1} - \hat{\lambda}_j)$ lie where the integrand $f(t) = t^{-\frac{1}{2}}(1-t)^{\frac{n-3}{2}}$ has a large derivative. Hence, although the effects of additional iterations on the accuracy can be assessed in terms of the integral $\mathcal{I}$, a qualitative interpretation seems difficult.

Just as the statistical analysis falls short in determining the preferred number of iterations, it fails regarding stopping and reorthogonalization criteria: even the simplest approximations of distributions become unwieldy [3, 14, 15], and probability density functions become dependent on the exact eigenvalues and on other assumptions that are difficult to verify [14]. Therefore, we did not extend the statistical analysis to the iterates.

7.4 Practical Considerations

The preceding statistical analysis qualitatively confirms the experimental observations regarding the starting vectors in Section 4. However, the analysis is

	for $n = 100$	for $n = 1000$	for $n = 10000$
$\sqrt{1-\epsilon^2}$	$P(\lvert\eta_i\rvert \geq \sqrt{1-\epsilon^2})$	$P(\lvert\eta_i\rvert \geq \sqrt{1-\epsilon^2})$	$P(\lvert\eta_i\rvert \geq \sqrt{1-\epsilon^2})$
$\leq 10^{-6}$	1.00 (16)	1.00 (16)	1.00 (16)
10^{-5}	1.00 (16)	0.99	0.90
10^{-4}	0.99	0.36	0.34
10^{-3}	0.92	0.33	0 (16)
10^{-2}	0.34	0 (2)	0 (16)
$\geq 10^{-1}$	0 (16)	0 (16)	0 (16)

Table 13: Lower bounds on the probability that $\eta_i^2 \geq 1-\epsilon^2$. Numbers in parentheses indicate the number of zero decimal places.

based on starting vectors with independent, normally distributed components, while the experiments were performed with uniformly distributed components in [-1,1] having some degree of dependence. Thus, the experimental starting vectors of length n are not uniformly distributed on the unit n-sphere but rather on an n-cube of height 2. Although normally distributed pseudorandom numbers can be generated at a higher cost than uniform ones [8], we will now show that uniform random numbers are acceptable substitutes.

The error in applying the analysis to uniformly distributed starting vectors may be estimated by circumscribing an n-sphere of radius $\sqrt{n}$ about the hypercube. A vector $y = (\eta_1, \ldots, \eta_n)^T = \sum_{i=1}^{n} \eta_i u_i$ on this sphere is a good approximation to a multiple $\sqrt{n}u_i$ of the eigenvector u_i if $|\eta_i| \geq \sqrt{n(1-\epsilon^2)}$. Following the derivation in Section 7.2, the probability of this occurrence is

$$P(\eta_i^2 \geq n(1-\epsilon^2)) = 1 - \alpha \int_0^{n(1-\epsilon^2)} t^{-\frac{1}{2}}(1-t)^{\frac{n-3}{2}} dt.$$

Lower bounds for this probability computed by Gauss-Legendre quadrature are given in Table 13.

The probability of a large component η_i in the uniform case is not as high as in the normally distributed case, but components with magnitude 10^{-6} can be expected with near certainty, and this value still suffices for fast convergence.

The linear independence of the pseudorandom starting vectors is demonstrated in Table 4.

8 Acknowledgements

The authors thank Stan Eisenstat and Jim Demmel for many helpful discussions.

References

[1] J. Barlow and J. Demmel, *Computing accurate eigensystems of scaled diagonally dominant matrices*, Report 421, Computer Science Dept, Courant Institute, 1988.

[2] H. Bowdler, R. Martin, and J. Wilkinson, *The QR and QL algorithms for symmetric matrices*, Numer. Math., 11 (1968), pp. 227–240.

[3] S. Crump, *The estimation of variance components in analysis of variance*, Biometrics, 2 (1946), pp. 7–11.

[4] J. Cuppen, *A divide and conquer method for the symmetric tridiagonal eigenproblem*, Numer. Math., 36 (1981), pp. 177–95.

[5] P. Deift, J. Demmel, L.-C. Li, and C. Tomei, *LAPACK working note #11: The bidiagonal singular value decomposition and Hamiltonian mechanics*, Computer Science Dept. Technical Report, Courant Institute, 1989.

[6] J. Demmel and W. Kahan, *LAPACK working note #3: Computing small singular values of bidiagonal matrices with guaranteed relative accuracy*, Mathematics and Computer Science Division, Argonne National Laboratory, 1988.

[7] J. Demmel and K. Veselić, *LAPACK working note #15: Jacobi's method is more accurate than QR*, Computer Science Dept. Technical Report, Courant Institute, 1989.

[8] L. Devroye, *Non-Uniform Random Variate Generation*, Springer-Verlag, 1986.

[9] J. Dixon, *Estimating extremal eigenvalues and condition numbers of matrices*, SIAM J. Numer. Anal., 20 (1983), pp. 812–814.

[10] J. Dongarra and D. Sorensen, *A fully parallel algorithm for the symmetric eigenvalue problem*, SIAM J. Sci. Stat. Comput., 8 (1987), pp. s139–s154.

[11] G. Golub and C. V. Loan, *Matrix Computations*, The Johns Hopkins Press, Baltimore, MD, 1983.

[12] R. Gregory and D. Karney, *A Collection of Matrices for Testing Computational Algorithms*, John Wiley and Sons, Inc., 1969.

[13] I. Ipsen and E. Jessup, *Solving the symmetric tridiagonal eigenvalue problem on the hypercube*, SIAM J. Sci. Stat. Comput., 11 (1990), pp. 203–229.

[14] E. Jessup, *Parallel Solution of the Symmetric Tridiagonal Eigenproblem*, PhD thesis, Dept of Computer Science, Yale University, 1989.

[15] N. Johnson and S. Kotz, *Continuous Univariate Distributions*, Houghton Mifflin Company, 1970.

[16] J. Kuczyński and H. Woźniakowski, *Estimating the largest eigenvalue by the power and Lanczos algorithms with a random start*, Technical Report, Dept. Computer Science, Columbia University, 1989.

[17] C. MOLER. Personal Communication, 1987.

[18] B. PARLETT, *The Symmetric Eigenvalue Problem*, Prentice Hall, Englewood Cliffs, NJ, 1980.

[19] B. SMITH, J. BOYLE, J. DONGARRA, B. GARBOW, Y. IKEBE, V. KLEMA, AND C. MOLER, *Matrix Eigensystem Routines–EISPACK Guide, Lecture Notes in Computer Science, Vol. 6, 2nd edition*, Springer-Verlag, 1976.

[20] J. WILKINSON, *Rounding Errors in Algebraic Processes*, Prentice-Hall, Inc., 1963.

[21] ——, *The Algebraic Eigenvalue Problem*, Clarendon Press, Oxford, 1965.

[22] ——, *Inverse iteration in theory and practice*, Symposia Mathematica Vol.X of the Institute Nationale di Alta Mathematica Monograf, Bologna, 19 (1972), pp. 361–379.