

Error Analysis of Reduction to Similar Banded Hessenberg Form

G. W. Howell* G. A. Geist† T. H. Rowan‡

July 16, 1998

Abstract

Gaussian similarity transformations can be used to alternately eliminate rows and columns to reduce a general real square matrix to a banded Hessenberg form. When multipliers are constrained to be small, the norm of the backward error is bounded by an expression involving overall conditioning of similarity transformations, size of multipliers, and maximal element size. Numerical testing with the functional stability analysis program INSTAB indicates good stability properties. Analysis and numerical experiments indicate that bounds on backward error increase smoothly with allowable multiplier size.

Key words. backward error analysis, eigenvalues, rounding error, similarity transformations, stability

AMS subject classifications. 15A23, 15A18, 65F15, 65F35

1 Introduction

A longstanding question has been whether a stable reduction to similar small-band form can be achieved. Recently, progress has been made in terms of look-ahead Lanczos methods and also by direct reduction by a sequence of similarity transformations. In both cases, a fundamental realization is that a stable reduction to strictly tridiagonal form is unlikely ([10], [11], [17], [20], [27]), with a backward error analysis [2] indicating the difficulties.

*Department of Applied Mathematics, Florida Institute of Technology, Melbourne, FL 32901 (howell@zach.fit.edu), (Author to whom correspondence should be addressed)

†Mathematical Sciences Section, Oak Ridge National Laboratory, P.O. Box 2008, Bldg. 6012, Oak Ridge, TN 37831-6367 (geist@msr.epm.ornl.gov)

‡Mathematical Sciences Section, Oak Ridge National Laboratory, P.O. Box 2008, Bldg. 6012, Oak Ridge, TN 37831-6367 (rowan@msr.epm.ornl.gov)

Many attempts have been made to improve stability by allowing a banded as opposed to a tridiagonal matrix. Direct methods that relax the tridiagonal constraint are proposed in [10], [11], [12], [23], and [27]. Look-ahead Lanczos methods have been implemented in [3], [7], [17], [20], and [27]. According to [2], there is as yet no backward error analysis of lookahead Lanczos methods.

The main purpose of this paper is to give a backward error analysis for the direct algorithm BHESSE proposed in [12]. Numeric examples given there illustrate that eigenvalues of the banded Hessenberg matrix produced by BHESSE differ only slightly from those produced by EISPACK or LAPACK. Here we analyze the BHESSE algorithm in terms of backward error and observed conditioning of the overall similarity transformations. We also provide a computable estimate of backward error and use the program INSTAB to illustrate the rarity of cases that exhibit large backward error.

A primary application of BHESSE is in determination of eigenvalues of sparse matrices. Suppose that a look-ahead Lanczos scheme produces a small-band Hessenberg matrix. Use of BHESSE as an iterative GR bulge-chasing (BR) iteration is often successful in computing the spectra of the small-band matrix in $O(n^2)$ operations and with $O(n)$ allocated storage [8]. Understanding the stability properties of BHESSE is a step toward understanding why eigenvalues computed by BR iteration are usually accurately determined.

The paper is organized as follows. Section 2 briefly describes the BHESSE procedure for reduction to small-band Hessenberg form. Section 3 gives a backward error analysis. Section 4 details efforts to use the functional stability analysis code INSTAB to search for an unstable case. Observed backward errors are discussed in Section 5. Section 6 briefly compares direct and Lanczos algorithms and describes some practical applications of the BHESSE algorithm.

2 Reduction to Banded Hessenberg Form

Reduction of a general matrix to a similar tridiagonal form is not likely to be a stable algorithm. After the (1,1) entry of the reduced form is fixed, similarity dictates the rest of the tridiagonal matrix (to diagonal scaling) [11]. Uniqueness in exact arithmetic breaks down when the matrix decomposes with a zero sub- or superdiagonal entry. Uniqueness and stability in rounding arithmetic degrades when the product of adjacent sub- and super-

diagonal elements is small, with the incidence of one or more small products increasingly likely as matrix size increases. The breakdown is exactly analogous to breakdown in the Lanczos method [27]. It is natural to attempt to improve stability by relaxing the constraint that a strictly tridiagonal matrix be returned, and instead constraining each similarity transformation to be well-conditioned.

We briefly describe here the BHES algorithm, deferring a discussion of its stability properties to the next sections. The first through $n - 2$ nd columns are successively eliminated below the subdiagonal by the following procedure. Suppose that $k - 1$ columns have been eliminated below the diagonal so that a matrix of the form

$$\begin{pmatrix} H_k & & 0 \\ & V^T & \\ 0 & u & G_{n-k} A_{n-k} G_{n-k}^{-1} \end{pmatrix} \quad (1)$$

has been obtained with H_k an upper Hessenberg matrix and where at least one row of V^T is nonzero. Then the k th column is eliminated below the subdiagonal by multiplication on the left by a Gaussian elementary transformation of the form

$$L_k = \text{diag}(I_{k+1}, \tilde{L}_{n-k-1}) = I_n - m_{k+1} e_{k+1}^T$$

where the first $k + 2$ entries of m_{k+1} are zero. Nonzero entries of the vector m_{k+1} are the necessary multipliers for the subdiagonal element to zero out the rest of the column. The inverse is $L_k^{-1} = I + m_{k+1} e_{k+1}^T$. After elimination of the k th column, any row v_i^T of V^T that has not previously been eliminated is eligible to have its last $n - k - 1$ entries zeroed by right multiplication with an elementary transformation of the form $R_k = \text{diag}(I_{k+1}, \tilde{R}_{n-k-1}) = I - e_{k+1} r_{k+1}^T$ with $R_k^{-1} = I + e_{k+1} r_{k+1}^T$. Elimination of one such (say the j th) row is made provided that j is the first row such that $\sec \theta / (n - k - 1) < \text{tol}$ where θ is the angle between the k th column and the j th row and tol is the user specified tolerance.

The essential step of the algorithm is as follows. Suppose that columns $1, \dots, k - 1$ have been eliminated and that H_k is of banded upper Hessenberg form, then for the k th step, consider the rows of V^T of Equation 1 where V is the matrix of column vectors (v_1, v_2, \dots, v_l) . If v_i^T is the first row such that

$$\frac{\|v_i^T\|_2 \|u\|_2}{(n - k - 1) |v_i^T u|} = \frac{\sec \theta}{n - k - 1} < \text{tol} \quad (2)$$

then v_i^T will be eliminated. If both a row and a column are to be eliminated then the k th column is first eliminated, followed by v_i^T , and BHES uses Gaussian transformations to eliminate both. Only one pivot is available for the row-column pair and it is chosen (in advance of zeroing the column) to minimize the maximal multiplier for the row-column pair. In BHES the first nonzero row satisfying Equation 2 is eliminated. If no row of V^T satisfies Equation 2, then no row is eliminated in conjunction with the k th column, and the pivot is chosen to minimize multipliers for the column.

Observation 2.1 *When the last entries of both a column u and a row v_i^T are eliminated, \sqrt{tol} corresponds to the root mean square of allowable multipliers. Note that $|v_i^T u| = \alpha\beta$, where α and β are the entries used to eliminate the rest of the column and row. Then from Equation 2*

$$\frac{\|v_i^T\|_2 \|u\|_2}{(n-k-1)|v_i^T u|} = \frac{\|\tilde{v}_i^T\|_2 \|\tilde{u}\|_2}{n-k-1} < tol$$

where $\tilde{v}_i^T = \pm v_i^T / \beta$ and $\tilde{u} = \pm u / \alpha$ are vectors with leading entries one and with the remaining entries the multipliers used for eliminating the row and column respectively. If $\|\tilde{v}_i^T\|_2 = \|\tilde{u}\|_2$ then the root mean square of each is bounded by \sqrt{tol} . The algorithm chooses the pivot to approximately equalize $\|\tilde{v}_i^T\|_\infty$ and $\|\tilde{u}\|_\infty$.

As is the case for Gaussian elimination or Gaussian reduction to full Hessenberg form, multipliers can be stored in the location that they were used to eliminate. If a pivot vector is also stored, then the original matrix can be reconstructed from the banded matrix by applying the similarity transformations in reverse.

BHES results in a banded Hessenberg form with bandwidth depending on the original matrix and tol . Ideally, we would like both a small bandwidth and a stable algorithm. In practice, there is a trade-off. A large value of tol results in a nearly tridiagonal matrix but large allowable multipliers and a less stable algorithm. Taking a flop as a paired addition and multiplication, BHES requires at most $4/3n^3$ flops compared to $5/6n^3$ flops for reduction to similar full Hessenberg form by Gaussian transformations or to $5/3n^3$ flops for reduction to Hessenberg form by Householder transformations.

The current Fortran 77 implementation of BHES performs four pairs of multiplications and additions on each column before operating on another column. This allows a row-column paired elimination to be performed while bringing the remaining part of the matrix into cache only once. This

version of BHESS runs slightly faster than the LAPACK Fortran 77 routine DGEHRD which reduces a matrix to Hessenberg form by Householder transformations. Those who have tuned BLAS-3 will find LAPACK DGEHRD to be significantly faster than the current version of BHESS.

A detailed description of the Hessenberg matrix returned by BHESS can be found in [12]. Numeric experiments given there indicate that in both well and poorly conditioned cases, loss of digits in computed eigenvalues from BHESS-QR is larger than but proportional to the loss of digits of the LAPACK algorithm. In the next sections we consider the extent to which BHESS is stable in a more general sense.

3 Stability of BHESS Reduction

If A is reduced to Hessenberg form \tilde{H} by an orthogonal similarity transformation, then

$$Q(A + E)Q^T = \tilde{H} \quad (3)$$

where

$$\|E\|_F / \|F\|_F \leq \phi(n)u$$

where ϕ is a low degree polynomial in maximal matrix dimension n (Wilkinson, [26, page 351]). If A is reduced to Hessenberg form \tilde{H} by Gaussian transformations with maximal column pivoting, then we have [26, page 364]

$$N^{-1}(A_0 - FN^{-1})N = \tilde{H} \quad (4)$$

where A_0 is a permutation of A and where $\|F\| = O(n) |A| u$ where $|A|$ is the maximal element of intermediate forms of A . If $|A|$ is small then F is satisfactorily small in norm, but backward error is guaranteed to be small only if $\|N^{-1}\|$ is also reasonably bounded. The EISPACK routine ELMHES uses maximal column pivoting for Gaussian reduction to similar Hessenberg form and is considered stable in practice.

The backward error of BHESS can be estimated from local backward errors occurring in elimination of a row-column pair and from the overall conditioning of the similarity transformations used for reduction. As in Wilkinson's analysis of reduction to to Hessenberg form, we simplify the analysis by considering the equivalent calculation for which all permutations are performed preliminary to eliminations. Lemma 3.1 shows that (as one might expect) elimination of a row-column pair by small multipliers produces

a small local backward error. Theorem 3.1 assembles the local backward errors into a global backward error.

Lemma 3.1 *Let A_k be the matrix produced by $k - 1$ steps of BHES. Then the first $k - 1$ columns of A_k are zero below the subdiagonal. Suppose that A_{k+1} is produced by zeroing the last $n - k - 2$ rows of the column and possibly the last $n - k - 2$ columns of the $j \leq k$ row of a matrix by Gaussian similarity transformations in floating point arithmetic with Gaussian multipliers satisfying 2. Let u be the machine precision, the largest number satisfying $1 = fl(1 + u)$. Then*

$$A_{k+1} = G_k^{-1}(A_k + E_k)G_k$$

where the local backward error $\|E_k\|$ satisfies

$$\|E_k\|_2 \leq |A_k| 12.08 \ n \ \|\tilde{m}_{k+1}\|_2^2 \ u \quad (5)$$

where $|A_k|$ is the largest absolute value over all elements of A_k and of intermediate values produced during the production of A_{k+1} from A_k and where \tilde{m}_{k+1} is a vector of length $n - k - 1$ with first entry 1 and the remaining entries the multipliers used to zero the k th column.

Proof. The proof is in the appendix.

Observation 3.1 *If row and column multipliers are equal in 2-norm, then $\|\tilde{m}_{k+1}\|_2^2 \leq (n - k) \text{ tol}$ and we have*

$$\|E_k\|_2 \leq 12.08 \ \text{tol} \ n(n - k) |A_k| u. \quad (6)$$

Applying Lemma 3.1 is straightforward.

Theorem 3.1 *Let*

$$H = fl(NAN^{-1})$$

be the Hessenberg matrix produced by the BHES algorithm in floating point arithmetic. Then

$$H = N(A_0 + E)N^{-1} \quad (7)$$

where

$$\|E\|_2 \leq 12.08 \ n \ \text{cond}_2(\tilde{N}) |A| \sum_{k=1}^{n-2} [\|\tilde{m}_{k+1}\|_2^2] \ u, \quad (8)$$

u the machine precision, and $A_0 = PAP^T$ is A with rows and columns permuted so that the reduction to banded Hessenberg form proceeds without permutations.

Proof. The overall backward error E can be given as

$$\begin{aligned} E &= E_1 + G_1 E_2 G_1^{-1} + G_1 G_2 E_3 G_2^{-1} G_1^{-1} + \dots \\ &\quad + G_1 G_2 \dots G_{n-2} E_{n-2} G_{n-2}^{-1} \dots G_2^{-1} G_1^{-1}. \end{aligned} \quad (9)$$

Denoting $\tilde{N}_k = G_1 G_2 \dots G_k$, take $\text{cond}_2(\tilde{N}) = \max_{k=1 \dots n-2} \text{cond}_2(\tilde{N}_k)$ and take $\| \cdot \|$ as the 2-norm. Then

$$\begin{aligned} \|E\| &\leq \|E_1\| + \|G_1\| \|E_2\| \|G_1^{-1}\| \\ &\quad + \|G_1 G_2\| \|E_3\| \|G_2^{-1} G_1^{-1}\| + \dots \\ &\quad + \|G_1 G_2 \dots G_{n-2}\| \|E_{n-2}\| \|G_{n-2}^{-1} \dots G_2^{-1} G_1^{-1}\| \\ &\leq \text{cond}_2(\tilde{N}) \left[\sum_{k=1}^{n-2} \|E_k\| \right]. \end{aligned} \quad (10)$$

Letting $|A| = \max_{k=1, \dots, n-2} |A_k|$ and using the result of Lemma 3.1, we have shown (8). QED.

If we make the obvious assumption that $\text{cond}_2(\tilde{N}_k)$ is monotone increasing with k , we can replace Equation 6 by

$$\|E\|_2 \leq \text{cond}_2(N) O(n) |A| \sum_{k=1}^{n-2} [\|\tilde{m}_{k+1}\|_2^2] u \quad (11)$$

where $N = \tilde{N}_{n-2}$ is the overall similarity transformation. BHES is seen to be backward stable on condition that $\|N\|$, $\|N^{-1}\|$, and $|A|$ are small. In practice, $\text{cond}(N)$ and $|A|$ are observable and tend not to be very large.

As in the case of LU decomposition, actual performance of the algorithm is somewhat better than the formal bound. A practical estimate on backward error seems to be

$$\|E\|_2 \leq n \sqrt{\text{cond}_2(N)} |H| u \quad (12)$$

where we take N as the overall similarity transformation and $|H|$ to be the observed maximal element in the reduced form. LINPACK or LAPACK estimators of $\text{cond}(N)$ allow computation of (12) with computational cost negligible compared to the overall cost of BHES reduction. Another reasonable and computable estimate is

$$\|E\|_2 \leq \text{tol}^2 |A| O(n^2) u. \quad (13)$$

The cases of LU decomposition of A and Gaussian reduction to Hessenberg form by bounded multipliers are similar in that both exhibit good numeric stability in most cases encountered. They are also similar in that cases can be devised with exponential growth of $|A|$ and the conditioning of L (or N). Businger [4] (1969) adapts Wilkinson's [26, p. 212] example for exponential growth of A and of the conditioning of $\text{cond}(L)$ in LU decomposition to show exponential growth of $|A|$ and $\text{cond}(N)$ in Gaussian reduction to similar Hessenberg form. Businger's examples also apply to the BHES algorithm.

As will be seen in the next two sections, conditioning of N in reduction to similar Hessenberg form is analogous to the conditioning of L in LU decomposition of A in that it is in practice rather rare to find examples for which N or L are poorly conditioned (Trefethen [21]). Similarly, element growth of A is modest in LU decomposition or Gaussian reduction to similar Hessenberg form. We conclude that the stability properties of Gaussian reduction to similar Hessenberg or banded Hessenberg form with bounded multipliers are similar to the stability properties of Gaussian elimination with bounded multipliers.

4 Experience with BHES and INSTAB

We tested the stability of BHES using the program INSTAB (Rowan [18]) to search for problems where instability is exhibited. This technique uses the relationship between the forward error, the backward error, and the problem's condition to estimate a lower bound on the backward error for any given problem. Since the forward error and the problem's condition can be estimated by treating the tested code as a black box, INSTAB can compute an estimate of the lower bound on the backward error by executing, as opposed to parsing, the code. The function that maps a problem's input to this estimate is maximized by a robust optimizer to search for examples of instability.

INSTAB has proved to be effective in practice, often discovering examples of instability on small problems within a few dozen runs of its optimizer. The functional stability analysis approach of INSTAB cannot prove stability by failing to find examples of instability, but an extensive optimization-based search can provide confidence in the stability of the algorithm.

The code INSTAB tested for instability with respect to the solution of the eigenvalue problem used BHES for the reduction to upper Hessenberg

form and the LAPACK routine DHSEQR to compute the eigenvalues. The approximate eigenvalues returned by the LAPACK routines DGEHRD and DHSEQR were assumed for the purpose of our testing to be the true eigenvalues. All of these routines were implemented in double precision.

INSTAB uses a simple problem model that assumes input and output are in the form of real vectors. INSTAB communicates with the tested software and with software capable of generating “true” answers through two interface routines. Each interface routine takes real vector input from INSTAB, represents it as actual parameters the software can use, and then represents the software’s output as a real vector that INSTAB can use to direct the search for instability.

Although a real $n \times n$ matrix is easily represented as a real n^2 -vector, representing a complex set (e.g. the eigenvalues returned by by BHES-DHSEQR) as a real vector is less straightforward. Given two complex sets and some vector distance measure, the imposed ordering on the set elements must yield vectors whose distance is an appropriate measure of the distance between the corresponding sets. The ordering of the elements of one set is arbitrary, but the second set must be ordered to minimize the distance between the vectors.

Finding such an optimal ordering is a network flow problem known as the *assignment problem*. Munkres’ algorithm [16] is one well known solution to the assignment problem, and Silver [19] has published an ALGOL implementation. For our tests we used a Fortran 77 translation of Silver’s implementation to obtain appropriate vector representations of the sets of complex eigenvalues.

For some fixed vector distance measure, let λ be the vector representation of the spectrum of A , using the element ordering returned by DGEHRD-DHSEQR. Similarly, let $\bar{\lambda}$ be the vector representation of the spectrum of \bar{A} , where \bar{A} is a perturbation to A . For our tests we use two vector distance measures for the output space,

1. the *vector* sense distance measure given by

$$d_v(\lambda, \bar{\lambda}) = \frac{\|\bar{\lambda} - \lambda\|}{\|\lambda\|}, \quad \|\lambda\| \neq 0,$$

and

2. the *component* sense distance measure given by

$$d_c(\lambda, \bar{\lambda}) = \|\sigma\|,$$

where

$$\sigma_i = \frac{|\bar{\lambda}_i - \lambda_i|}{|\lambda_i|}, \quad |\lambda_i| \neq 0.$$

The function d_v is simply the relative error of a vector, whereas d_c is the vector of relative errors in each component. We tested stability both in the vector sense and in the component sense. These stability senses correspond roughly to *normwise backward stability* and *componentwise backward stability*, as described in the LAPACK User's Guide [1]. With these distance measures defined, we can now describe the steps INSTAB takes to search for instability.

INSTAB estimates $C(A)$, the condition of computing the eigenvalues of A , by

$$C^*(A) = \max_k \frac{d(\lambda, \bar{\lambda}^{(k)})}{d_v(A, \bar{A}^{(k)})}$$

where k is the number of experimental perturbations to A , d is d_v or d_c depending on whether stability is being measured in the vector or the component sense, and $\bar{A}^{(k)}$ and $\bar{\lambda}^{(k)}$ are a perturbation to A and its eigenvalues. Using the Euclidean norm as in our experiments and taking into account the representation of A as a vector, this condition estimate for the vector sense analysis becomes

$$C^*(A) = \max_k \frac{\|\bar{\lambda}^{(k)} - \lambda\|_2 / \|\lambda\|_2}{\|\bar{A}^{(k)} - A\|_F / \|A\|_F}.$$

Let λ^* be the vector representation of the spectrum computed by BHES-DHSEQR, using the element ordering that minimizes the vector distance to λ . INSTAB estimates $F(A)$, the forward error in the computed eigenvalues of A , by

$$F^*(A) = d(\lambda, \lambda^*).$$

The ratio of the estimates to $C(A)$ and $F(A)$ give B_L^* , an estimated lower bound on the backward error [18]:

$$B_L^*(A) = \frac{F^*(A)}{C^*(A)}.$$

INSTAB maximizes $B_L^*(A)$ over the space of matrices. Any large values found would be an indication of instability in BHES-DHSEQR.

We tested for instability under the parameter configurations $n = 5, 10, 15$ and $tol = 1, 10, 100$. We ran 100 optimizations, starting from randomly

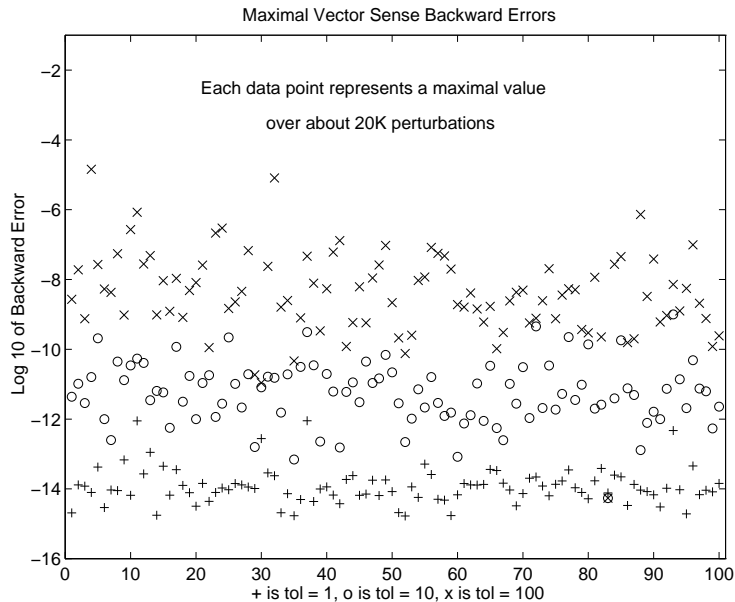


Figure 1: 100 INSTAB searches for large vector-sense backward errors for $n = 15$, and $tol = 1, 10$, and 100 . Each data point represents a maximal backward error found in examination of over 20K eigenproblems.

generated problems with entries of the matrix from a uniform distribution bounded by -1 and 1 , for each of these 18 configurations. The problem size was kept small because each optimization requires roughly $100n^2$ executions of BHES-DHSEQR. Small problem size may not be a serious limitation in drawing general conclusions about stability properties of an algorithm because instability, if it exists, often manifests itself in small problems produced by the optimization procedure of INSTAB.

Figure 1 summarizes the *vector sense* stability tests for $n = 15$. For $tol = 1$, the *vector sense* backward error estimates are a small multiple of the machine precision (approximately 10^{-16}). For these cases, therefore, BHES exhibited no evidence of instability in the *vector* sense.

As expected, increasing tol increased the instability of the algorithm. Loss of stability seems fairly regular with increasing tol . Note that for $tol = 100$ backward error estimates greater than 10^{-8} translate to instability sufficient to cause worse than single precision results on well-conditioned problems.

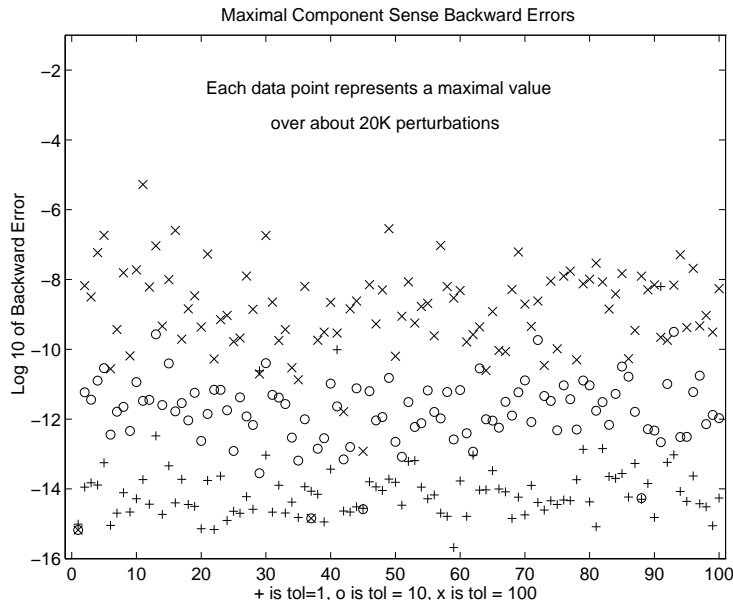


Figure 2: 100 INSTAB searches for large component-sense backward errors for $n = 15$, and $tol = 1, 10$, and 100 . Each data point represents a maximal backward error found in examination of over 20K eigenproblems.

To further study the effects of increasing n on the algorithm’s vector-sense stability, we also ran tests with $n = 30$ and $tol = 1$. Backward error estimates from approximately 600K eigenproblems appear comparable to those with the smaller values of n . However, tests covering only a few small values of n are insufficient for drawing general conclusions about the effect increasing n has on algorithm stability.

Figure 2 summarizes the component-sense stability tests for $n = 15$. Although component-sense stability is a stronger type of stability, most tests had results comparable to vector-sense tests. In particular, for $tol = 1$ and $n = 15$ only three estimates of a hundred were significantly higher than a small multiple of the machine precision. This indicates that on well-conditioned problems BHESS will usually return eigenvalues of high relative accuracy. Many of the computed forward errors were comparatively large, indicating that the initially well-conditioned problems were perturbed to be poorly conditioned in the course of the optimization.

These tests appear to provide strong evidence of BHESS’s stability when

small tolerances on the multipliers are used. Further, allowing larger (but bounded) multipliers does not appear to cause a sudden catastrophic loss of stability.

5 Numerical Observations of Backward Stability

In previous work (Howell, Geist, Diaa [12]), we described a number of numerical experiments in which a Fortran 77 version of BHES followed by QR iteration returned very nearly the same eigenvalues as the EISPACK routine ELMHES followed by QR iteration. The experiments discussed here are with a MATLAB version of BHES. The MATLAB version estimates not only the extent to which application of BHES preserves eigenvalues, but also computes backward error, and other quantities such as overall conditioning of the similarity transformations and observed element growth.

MATLAB backward errors were computed in several ways. Assume H is the banded Hessenberg matrix produced by BHES from A . A first approach is to apply in reverse the Gaussian transformations used by BHES to convert H back to an approximate \tilde{A} . We then take the backward error as $E = \tilde{A} - A$ and estimate its size as $\|E\|/\|A\|$

We also estimated backward error by the following method. We transformed A to banded Hessenberg H , saving multipliers, took random entries between minus one and one in all nonzero locations of H , applied a backward similarity transformation, added uniformly distributed random entries equally likely to be positive or negative to current nonzero locations, applied the next backward similarity transformation, etc. The resulting backward error matrix was then normalized by multiplying by $u = 10^{-16}$ and the backward error given as $\|E\|/\|A\|$. Computed backward errors by the second method were somewhat larger than those from the first method but quite comparable.

In Figure 3, we plot MATLAB computed conditioning of $\|N\|$ when BHES is applied to matrices with entries drawn from a uniform distribution between -1 and 1, taking the variance of multipliers to be bounded by 1 ($tol = 1$) and 100 ($tol = 100$) respectively. For either case, the slope of the curve on the log-log plot is about two, indicating that the $cond_2(N)$ increases proportionally to n^2 .

In Figure 4, computed backward error $A - NHN^{-1}$ is plotted for $tol = 1$ and $tol = 100$ and goes as the square of matrix size. Typically, increasing tol allows backward error and condition numbers to increase.

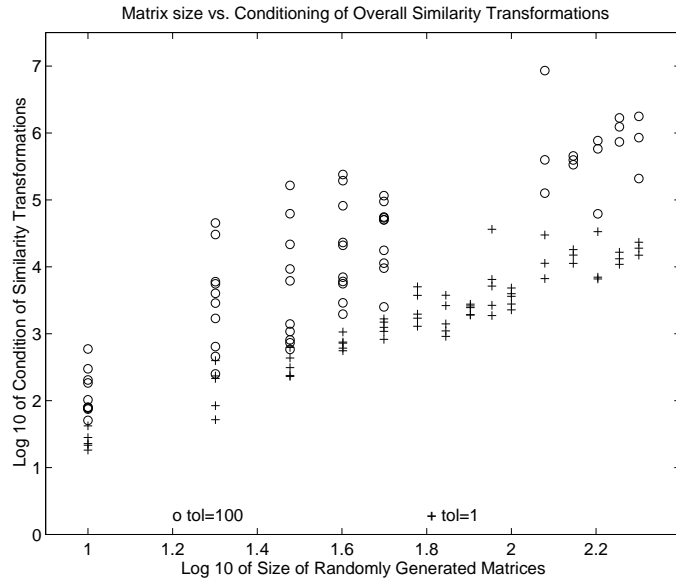


Figure 3: Conditioning of Similarity Transformations

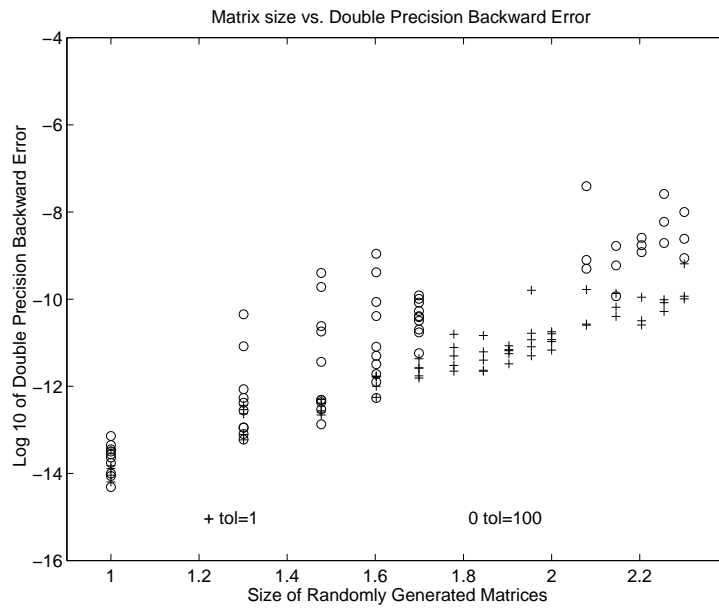


Figure 4: Computed Backward Error

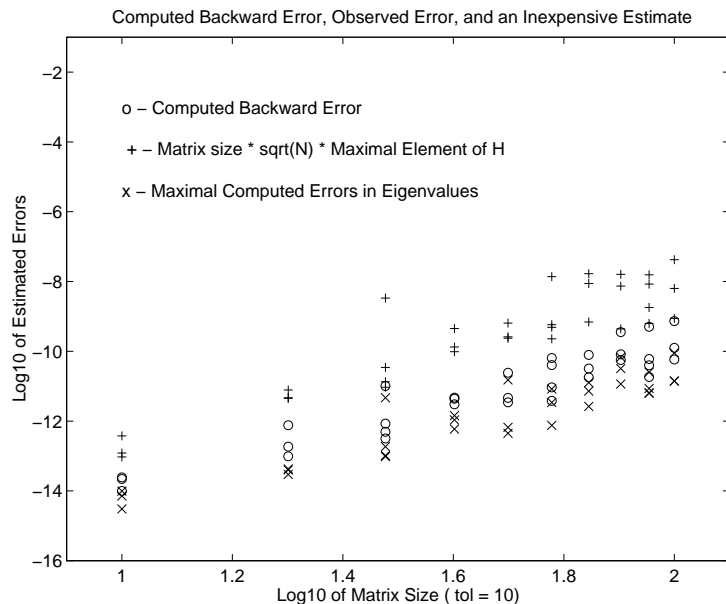


Figure 5: Error, Backward Error, and an Inexpensive Estimate

Figure 5 plots the maximal difference between eigenvalues as computed by MATLAB from the original matrix and eigenvalues computed by MATLAB from a matrix returned by BHES. It also plots backward errors and the the error estimate (Equation 12)

$$e_n = n\sqrt{\text{cond}_2(N)}|H|u$$

where $|H|$ is the largest entry in absolute value of the matrix returned by BHES and u is the machine precision. e_n is seen to be (at least in this case) a good predictor of the maximal eigenvalue error. A more sophisticated estimator of accuracy could also take into account eigenvalue condition number. Since the matrix returned by BHES is low bandwidth, Rayleigh quotient iteration usually returns refined eigenvalues and eigenvectors and their condition numbers quickly and accurately. If condition of the transformation is estimated by LAPACK or LINPACK, e_n can be estimated in $O(n^2)$ operations. Comparable plots to 3 and 4 for Gaussian elimination with threshold pivoting are given in [21].

The Businger [4] examples show that as in the case of Gaussian elimination, examples exist for which BHES exhibits large backward error through

exponential growth of conditioning and/or element size. Such examples do not seem to be common. For instance, we tried several different sizes for each of the available test matrices from Higham’s MATLAB Matrix toolbox [9] and highly nonnormal matrices suggested by S. Lee [13]. None of Higham’s or Lee’s examples caused large backward error for BHES.

6 Comparison to Lanczos methods and Applications

We have presented a form of direct reduction to small-band Hessenberg form and explored its stability properties. Direct reduction to small-band form has a good deal in common with Lanczos methods, which involve a three-term recurrence of biorthogonal vectors. See for example, Parlett [17], Freund, Gutknecht, and Nachtigal [7], Ye [27], Bai, Day, and Ye [3]. Advantages to the Lanczos method for sparse matrices are that the original matrix is accessed only to form a matrix vector product, thus there is no “fill-in”. Also, if only a three term recurrence is used, then previous vectors can be discarded, minimizing storage.

Reduction to tridiagonal form shares the potential for breakdown and poor conditioning with Lanczos methods. In many look-ahead Lanczos methods, 2×2 , or more generally $k \times k$ pivots are used to restore the three-term recurrence (tridiagonal bandwidth) after a $k \times k$ bump has been formed [17], [7]. Conditioning for a Lanczos step may be allowed to be as high as \sqrt{u} where u is the machine precision. In contrast, the Lanczos variant due to Ye [27] and also BHES can enforce a much more stringent control on conditioning of a given step but pay a price of relatively wide bandwidths. In the Lanczos method, the widened bandwidth corresponds to a longer recurrence.

Some explicit comparisons between Ye’s algorithm and direct algorithms can be made. The tolerance used by Ye when determining breakdown is the same as that used in Howell ([1994, [10]) in the preliminary version of BHES, i.e., a tolerance on the cosine of the angle between two vectors. Also the form of the banded Hessenberg matrix returned by Ye’s algorithm is the same as the form of the banded Hessenberg matrix in [10], for which only the topmost nonzero row is eliminated with a given column elimination. Two differences of BHES are that tolerance is adjusted for length of the current row, and that since more rows than the topmost are eligible for elimination the resulting matrix has a more jagged profile. In numeric tests,

the algorithm of [10] often gives bandwidth proportional to n while BHES gives numerically more stable performance with bandwidth $O(\sqrt{n})$ on the same cases.

In contrast to the well-behaved vectors of multipliers of BHES, the locally biorthogonal vectors produced by Lanczos methods tend to become very nearly linearly dependent over twenty or thirty steps. The available explicit backward error analysis indicates difficulties [2] with the unsymmetric (non-lookahead) Lanczos algorithm. Storage and reorthogonalization of Lanczos vectors is sometimes combined with look-ahead techniques to improve the accuracy of the computation [3].

Both BHES and look-ahead Lanczos methods reduce the problem of finding eigenvalues of a general matrix to that of finding eigenvalues of a small-band matrix. In either case, we are left with the question of finding the eigenvalues of such a matrix.

BHES can iteratively chase bulges of the small-band Hessenberg matrix in an implicit GR or so-called BR iteration [8]. Well-known GR iterations are QR and LR with pivoting [25, 24]. Unfortunately, QR or LR iteration with pivoting converts a small-band Hessenberg matrix to full Hessenberg form. LR without pivoting preserves the small-band Hessenberg form, but is unstable to the extent that (in our experience with using exceptional shifts to attempt to avoid poor multipliers) arithmetic of precision higher than double is needed for most problems of size greater than 400.

Using BHES iteratively as a GR iteration constricts bandwidth, allowing (often) a speedy computation with bounded multipliers [8]. For example, we applied look-ahead Lanczos to a Tolosa matrix of size 20K to form a 10K small-band matrix and determined eigenvalues of the small-band matrix on a DEC workstation. QR determination of eigenvalues of this matrix would require 800MB of memory and about 60 times as many computations as iterative BHES (BR iteration).

BR iteration can also be used on the matrices returned by BHES. In this case, the reduction costs $O(n^3)$ computations, so that overall computational costs are proportional to those of LAPACK. Since the computation with BHES is inherently somewhat less accurate than LAPACK, the usefulness of BHES is not as clear cut. For BR iteration to be effective, small bandwidths ($tol \geq 15$ for BHES) are typically required. For such high values of tol , accuracy of the spectra suffers but the refinement scheme [5] can be used to obtain higher accuracy. Users who do not have tuned BLAS-3 will find that BHES-BR is about twice as fast as LAPACK. Users with access to tuned BLAS-3 will enjoy rather less speedup.

L. Zeng proposes to use BHES with lower tolerances to get a relatively fat matrix for a homotopy based method of finding a spectra [15], via Newton's method on the determinant of the matrix. The cost of iteration is proportional to the number of nonzeros in the Hessenberg matrix. Potentially, the use of low *tol* for BHES followed by a homotopy method offers nearly the accuracy of LAPACK with a substantial speedup.

Another possible application is to ADI solution of the Sylvester equation $AX + XB = C$. If A and B are reduced to similar small-band Hessenberg forms \tilde{A} and \tilde{B} , we get an equivalent equation $\tilde{A}Y + Y\tilde{B} = \tilde{C}$, where ADI iterations on the reduced forms are relatively inexpensive so that the main expense is in the conversion to reduced form. See Levenberg and Reichel [14], Ellner and Wachspress [6], and Wachspress [22].

Acknowledgments

We wish to acknowledge support under the ORISE/ORAU summer faculty research program for Gary Howell in 1994 and 1995. We thank E. D'Azevedo, J. Dongarra, C. Romine, E. Wachspress and D. S. Watkins for advice, ideas, and encouragement.

Appendix

Proof of Lemma 3.1. If both a row and column elimination are performed A_{k+1} is produced from A_k as

$$\begin{aligned} A_{k+1} &= fl(R_k^{-1}L_k A_k L_k^{-1}R_k) \\ &= R_k^{-1}L_k[A_k + E_k^{(1)} + L_k^{-1}E_k^{(2)}L_k]L_k^{-1}R_k \end{aligned}$$

where $E_k^{(1)}$ satisfies

$$A_k^{(2)} = L_k^{-1}(A_k + E_k^{(1)})L_k$$

and $E_k^{(2)}$ satisfies

$$A_{k+1} = R_k(A_k^{(2)} + E_k^{(2)})R_k^{-1}.$$

If no row elimination is performed we can consider $E_k^{(2)}$ to be the zero matrix. In words, $E_k^{(1)}$ is the backward error matrix of the similarity transformation eliminating the k th column below the subdiagonal and $E_k^{(2)}$ is the backward

error matrix for the similarity transformation eliminating the last $n - k - 2$ entries of some row m , $m \leq k$. Then

$$E_k = E_k^{(1)} + L_k^{-1} E_k^{(2)} L_k$$

and

$$\|E_k\|_2 \leq \|E_k^{(1)}\|_2 + \|L_k^{-1}\|_2 \|E_k^{(2)}\|_2 \|L_k\|_2.$$

We are assuming that L_k is an elementary Gaussian matrix of the form $I_n + m_{k+1} e_{k+1}^T$ where the n -vector m_{k+1} has a leading $k + 2$ zeros so that L_k is an identity matrix plus a column of multipliers in the $k + 1$ st column below the diagonal. Then

$$L_k^T L_k = \begin{pmatrix} I_k(1 + \sum_j m_j^2) & 0 \\ 0 & m_{k+1}^T \\ & & I_{n-k-1} \end{pmatrix} \quad (14)$$

Using diagonal rescaling to convert the $k + 1$ st column into ones, the $k + 1$ st row has each element squared. By Gershgorin's Theorem, the largest eigenvalue associated with the $k + 1$ st diagonal entry is bounded by $1 + 2 \sum_j m_j^2$. Thus

$$\|L_k\|_2 \leq (1 + 2m_{k+1}^T m_{k+1})^{1/2}.$$

Since $L_k^{-1} = I_n - m_{k+1} e_{k+1}^T$, we also have $\|L_k^{-1}\|_2 \leq (1 + 2m_{k+1}^T m_{k+1})^{1/2}$. Taking $\tilde{m}_{k+1} = [1, m_{k+1}^T]$ gives

$$\text{cond}_2(L_k) \leq 2\|\tilde{m}_{k+1}\|_2^2 - 1.$$

We estimate $\|E_k^{(1)}\|_2$. Take

$$A_k^{(1)} = fl(L_k A_k) \quad (15)$$

and

$$A_k^{(2)} = fl(A_k^{(1)} L_k^{-1}) = (A_k^{(1)} + E_k^{(1,1)}) L_k^{-1} \quad (16)$$

where producing $A_k^{(2)}$ from $A_k^{(1)}$ is performed by adding multiples of the successive columns to the $k + 1$ st column of $A_k^{(1)}$, i.e.,

$$a_{i,k+1}^{(k,2)} = fl(a_{i,k+1}^{(k,1)} + \sum_{j=k+2}^n m_{jk} a_{ij}^{(k,1)}). \quad (17)$$

Then

$$\begin{aligned} fl(a_{i,k+1}^{(k,1)} + m_{k+2,k} a_{i,k+2}^{(k,1)}) \\ = [a_{i,k+1} + m_{k+2,k} a_{i,k+2}^{(k,1)} (1 + \epsilon_{i,k+2}^{(k,2)})] (1 + \eta_{i,k+2}^{(k,2)}). \end{aligned} \quad (18)$$

and for $l = k + 3, \dots, n$

$$\begin{aligned} fl(a_{i,k+1}^{(k,1)} + \sum_{j=k+2}^l m_{jk} a_{ij}^{(k,1)}) = \\ [fl(a_{i,k+1}^{(k,1)} + \sum_{j=k+2}^{l-1} m_{jk} a_{ij}^{(k,1)}) + m_{lk} a_{il}^{(k,1)} (1 + \epsilon_{il}^{(k,2)})] (1 + \eta_{il}^{(k,2)}) \end{aligned} \quad (19)$$

For the l th entry $e_{il}^{(k,1)}$ of the i th row of $E_k^{(1,1)}$ we can take $|\eta_{il}^{(k,2)}|, |\epsilon_{il}^{(k,2)}| < u$ and thus assign

$$|e_{il}^{(k,1)}| \leq \{ |fl(a_{i,k+1}^{(k,1)} + \sum_{j=k+2}^{l-1} m_{jk} a_{ij}^{(k,1)})| + 2|m_{lk} a_{il}^{(k,1)}| \} (1.01 - u). \quad (20)$$

Define

$$|a^{(k,2)}| = \max \{ |fl(a_{i,k+1}^{(k,1)} + \sum_{j=k+2}^{l-1} m_{jk} a_{ij}^{(k,1)})|, |m_{lk} a_{il}^{(k,1)}| \}. \quad (21)$$

Then

$$|e_{il}^{(k,1)}| \leq 3.03 |a^{(k,2)}| u. \quad (22)$$

We want

$$A_k^{(1)} = fl(L_k A_k)$$

and also that

$$A_k^{(1)} + E_k^{(1,1)} = L_k (A_k + E_k^{(1)}).$$

Denoting entries of $E_k^{(1,1)}$ as $e_{il}^{(k,1)}$ and of $E_k^{(1)}$ as $e_{il}^{(k)}$ then for $j = k + 1, \dots, n$ and $i = k + 2, \dots, n$

$$fl(a_{ij}^{(k)} - m_{ik} a_{k+1,j}^{(k)} + e_{il}^{(k,1)}) = (a_{ij}^{(k)} + e_{ij}^{(k)}) - m_{ik} (a_{k+1,j}^{(k)} + e_{k+1,j}^{(k)}) \quad (23)$$

or on converting the expression on the left to an equivalent expression in exact arithmetic

$$\begin{aligned} [a_{ij}^{(k)} - m_{ik} a_{k+1,j}^{(k)} (1 + \epsilon_{ij}^{(k,1)})] (1 + \eta_{ij}^{(k,1)}) + e_{ij}^{(k,1)} = \\ (a_{ij}^{(k)} + e_{ij}^{(k)}) - m_{ik} (a_{k+1,j}^{(k)} + e_{k+1,j}^{(k)}) \end{aligned} \quad (24)$$

where

$$m_{ik} = (a_{ik}^{(k)} / a_{k+1,k}^{(k)}) (1 + \xi_i^{(k)}) \quad (25)$$

so that the exact zeroing of the k th column requires a backward error $e_{ik}^{(k)} = \xi_i^{(k)} a_{k+1,k}^{(k)}$ for the (i, k) entry. We have $|\epsilon_{ij}^{(k)}|, |\eta_{ij}^{(k)}|, |\xi_i^{(k)}| < u$. Assigning $e_{k+1,j}^{(k)}$ to be zero, we can uniquely determine the rest of the backward error matrix $E_k^{(1)}$ by

$$e_{ij}^{(k)} = \eta_{ij}^{(k,1)} a_{ij}^{(k)} - m_{ik} a_{k+1,j}^{(k)} (\eta_{ij}^{(k,1)} + \epsilon_{ij}^{(k,1)} + \eta_{ij}^{(k,1)} \epsilon_{ij}^{(k,1)}) + e_{ij}^{(k,1)}. \quad (26)$$

Recall again that $A_k^{(1)} = fl(L_k A_k)$ and denote

$$|a^{(k,1)}| = \max\{|a_{ij}^{(k)}|, |m_{ik} a_{k+1,j}^{(k)}|\}.$$

Then for rows $k+2$ to n and columns $k+1$ to n

$$\begin{aligned} |e_{ij}^{(k)}| &\leq (3.01) u |a^{(k,1)}| + |e_{ij}^{(k,1)}| \\ &\leq \{(3.01)|a^{(k,1)}| + (3.03)|a^{(k,2)}|\}u \end{aligned} \quad (27)$$

so that backward errors from multiplication on the left by L_k and on the right by L_k^{-1} are seen to be additive and of roughly the same size. Take $|A_k^{(1)}| = \max\{|a^{(k,1)}|, |a^{(k,2)}|\}$. Since the entries of $E_k^{(1)}$ are bounded by 6.04 $|A_k^{(1)}| u$, it is not hard to see that

$$\|E_k^{(1)}\|_2 \leq 6.04 n |A_k^{(1)}| u.$$

Recasting the same arguments gives

$$\|E_k^{(2)}\|_2 \leq 6.04 n |A_k^{(2)}| u$$

with $|A_k^{(2)}| = \max\{|a^{(k,3)}|, |a^{(k,4)}|\}$. Take

$$|A_k| = \max\{|A_k^{(1)}|, |A_k^{(2)}|\}.$$

Recalling that $E_k = E_k^{(1)} + L_k^{-1} E_k^{(2)} L_k$, we have

$$\begin{aligned} \|E_k\|_2 &\leq \|E_k^{(1)}\|_2 + \|L_k^{-1}\|_2 \|E_k^{(2)}\|_2 \|L_k\|_2 \\ &\leq 6.04 n |A_k| u + 6.04 n |A_k| u (2\|\tilde{m}_{k+1}\|_2^2 - 1) \\ &\leq 12.08 n |A_k| \|\tilde{m}_{k+1}\|_2^2. \end{aligned} \quad (28)$$

This concludes the proof of Lemma 3.1.

References

- [1] E. Anderson, Z. Bai, C. Bischof, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, S. Ostrouchov, and D. Sorensen. *LAPACK Users' Guide*. SIAM, Philadelphia, 1992.
- [2] Z. Bai. Error analysis of the Lanczos algorithm for the nonsymmetric eigenvalue problem. *Math. Comp.*, 62(205):209–226, 1994.
- [3] Z. Bai, D. Day, and Q. Ye. ABLE: An adaptive block Lanczos method for non-Hermitian eigenvalue problems. Research Report 95-04, Department of Mathematics, University of Kentucky, 1995.
- [4] P. A. Businger. Reducing a matrix to Hessenberg form. *Math. Comp.*, 23:819–921, 1969.
- [5] J.J. Dongarra, G.A. Geist, , and C.H. Romine. Algorithm 710: FORTRAN subroutines for computing the eigenvalues and eigenvectors of a general matrix by reduction to general tridiagonal form. *ACM Trans. Math. Software*, pages 392–400, 1992.
- [6] N. S. Ellner and E. L. Wachspress. Alternating direction implicit iteration for systems with complex spectra. *SIAM J. Num. Anal.*, 28:859–870, 1991.
- [7] Roland W. Freund, M. H. Gutknecht, and N. M. Nachtigal. An implementation of the look-ahead Lanczos algorithm for non-Hermitian matrices. *SIAM J. Sci. Computing*, 14(1):137–158, 1993.
- [8] G. A. Geist, G. W. Howell, and D. S. Watkins. The BR eigenvalue algorithm. submitted to the special edition of SIAM J. Matrix Anal. Appl. for the Sparse Matrix Conference at Cour d'Alene, 1996, available from D. S. Watkins at watkins@wsu.edu.
- [9] N. J. Higham. Algorithm 694: A collection of test matrices in matlab. *ACM Trans. Math. Software*, 17(3):289–305, September 1991.
- [10] G. W. Howell. Efficient computation of eigenvalues of randomly generated matrices. *J. Applied Math. and Comp.*, 66:9–24, 1994.
- [11] G. W. Howell and G. A. Geist. Direct reduction to a similar near-tridiagonal form. In *Proc. of the ISCA 8th International Conference on*

- Parallel and Distributed Computing Systems*, pages 426–432, Raleigh, NC, 1995. International Society Computer and Applications.
- [12] G. W. Howell, G. A. Geist, and N. Diao. Gaussian reduction to a near-tridiagonal Hessenberg form: Algorithm BHES. submitted to ACM Trans. Math. Software in April 1994, available from the first author at howell@zach.fit.edu.
 - [13] S. L. Lee. A practical upper bound for departure from normality. *SIAM J. Matrix Anal. Appl.*, 16(2):462–468, 1995.
 - [14] N. Levenberg and L. Reichel. A generalized ADI iterative method. *Num. Math.*, 66:215–233, 1992.
 - [15] T. Y. Li and Z. Zeng. Homotopy-determinant algorithm for solving the nonsymmetric eigenvalue problem. *Math. Comp.*, 1994.
 - [16] J. Munkres. Algorithms for the assignment and transportation problems. *J. SIAM*, 5:32–38, 1957.
 - [17] B. N. Parlett. Reduction to tridiagonal form and minimal realizations. *SIAM J. Matrix Anal. Appl.*, 13(2):567–593, 1992.
 - [18] T. H. Rowan. *Functional Stability Analysis of Numerical Algorithms*. Ph.d. thesis, Department of Computer Sciences, University of Texas at Austin, Austin, TX, 1990. <http://www.epm.ornl.gov/~rowan/thesis/>.
 - [19] R. Silver. An algorithm for the assignment problem. *Comm. ACM*, 3:605–606, 1960.
 - [20] D. R. Taylor. *Analysis of the look ahead Lanczos algorithm*. Ph.d. thesis, University of California, Berkeley, Berkeley, CA, 1982.
 - [21] L. N. Trefethen and R. S. Schreiber. Average-case stability of Gaussian elimination. *SIAM J. Matrix Anal. Appl.*, 11:335–360, 1990.
 - [22] E. L. Wachspress. The ADI model problem. self-published monograph, Windsor, CA, 1995.
 - [23] E. L. Wachspress. Similarity matrix reduction to banded form. manuscript, 1995.
 - [24] D. S. Watkins and L. Elsner. Chasing algorithms for the eigenvalue problem. *SIAM J. Matrix Anal. Appl.*, 12(2):374–384, 1991.

- [25] D. S. Watkins and L. Elsner. Convergence of algorithms of decomposition type for the eigenvalue problem. *Lin. Alg. and Applic*, 143:19–47, 1991.
- [26] J. H. Wilkinson. *The Algebraic Eigenvalue Problem*. Clarendon Press, Oxford, England, 1965.
- [27] Q. Ye. A breakdown-free variation of the nonsymmetric Lanczos algorithms. *Math. Comp.*, 62:179–207, 1994.