ORNL/TM-11089

OAK RIDGE
NATIONAL
LABORATORY

MARTIN MARIETTA

Stabilized Gaussian Reduction of an Arbitrary Matrix to Tridiagonal Form

G. A. Geist
A. Lu
E. L. Wachspress

OPERATED BY
MARTIN MARIETTA ENERGY SYSTEMS, INC.
FOR THE UNITED STATES
DEPARTMENT OF ENERGY

ORNL/TM-11089

Engineering Physics and Mathematics Division

Mathematical Sciences Section

STABILIZED GAUSSIAN REDUCTION OF AN ARBITRARY MATRIX TO TRIDIAGONAL FORM

G. A. Geist *
A. Lu **
E. L. Wachspress **

* Oak Ridge National Laboratory
Mathematical Sciences Section
P.O. Box 2009, Bldg. 9207-A
Oak Ridge, TN 37831-8083

** University of Tennessee
Department of Mathematics
Knoxville, TN 37996-1300

Date Published: June, 1989

Research was supported by the Applied Mathematical Sciences Research Program of the Office of Energy Research, U.S. Department of Energy and the Science Alliance of the State of Tennessee.

Prepared by the
Oak Ridge National Laboratory
Oak Ridge, Tennessee 37831
operated by
Martin Marietta Energy Systems, Inc.
for the
U.S. DEPARTMENT OF ENERGY
under Contract No. DE-AC-05-84OR21400

Contents

STABILIZED GAUSSIAN REDUCTION OF AN ARBITRARY MATRIX TO TRIDIAGONAL FORM

G. A. Geist

A. Lu

E. L. Wachspress

Abstract

This report presents several ideas for improving the stability of Gaussian reduction of an arbitrary real matrix to tridiagonal form. First, we analyze conditions under which reduction algorithms break down or become unstable. Second, we discuss how methods of threshold pivoting decrease the probability of these conditions occuring. Finally, we present new methods for recovering from breakdown when it does occur. The class of matrices that can be successfully reduced is significantly broadened by these new recovery algorithms.

1. Introduction

The reduction of a general dense $n \times n$ matrix A to a similar tridiagonal matrix is a difficult task that was studied intensively several years ago [7,11]. The most straightforward approach is to reduce A to upper Hessenberg form H and then apply elementary similarity transformations to H to reduce it to tridiagonal form S. Unfortunately, large multipliers (small pivots) are encountered for some matrices, and the reduction suffers from instabilities or breaks down entirely [9]. Wilkinson [16] discussed this situation in some detail and suggested possible remedies, none of which appeared adequate to him.

Our initial interest in reduction to tridiagonal form grew from the work of Wachspress [14], who was solving the Lyapunov matrix equation, $AX + XA^T = C$. He observed that if A could be reduced to tridiagonal form, S, then the equation $SZ + ZS^T = C_s$ could be solved by ADI iteration in $O(n^2)$ flops [13,12]. Matrix C_s is derived from C during the reduction of A to S, and X is recovered from Z by the inverse of this reduction. The entire computation requires about $5n^3$ flops as compared with about $15n^3$ flops for the standard method of Bartels-Stewart [1].

Computation of all the eigenvalues of a real nonsymmetric $n \times n$ matrix is one of the more demanding tasks in numerical linear algebra. The standard method, which is robust and stable, involves reduction to Hessenberg form by Householder transformations followed by reduction to real Schur form by a sequence of shifted QR iterations [10]. The Hessenberg reduction requires $\frac{5}{3}n^3$ flops. Experience with QR indicates that an average of two iterations are required to find each eigenvalue. As each eigenvalue is found, the size of the remaining submatrix decreases by one. Each QR iteration on a matrix of order k requires $5k^2$ flops, so that, as k runs from 1 to n, a total of $\frac{10}{3}n^3$ flops are needed for the QR reduction to Schur form. Thus, the Hessenberg and Schur reductions require a total of about $5n^3$ flops. On the other hand, if the matrix can be reduced to tridiagonal form, then the QR iterations may be replaced by LR iterations, which, unlike QR, preserve the tridiagonal form. The implicit double-shift LR iteration applied to a tridiagonal matrix requires only $6k$ flops per iteration for a matrix of order k and commonly takes less than four iterations per eigenvalue for a total flop-count of $12n^2$. A reliable method of reducing a matrix to tridiagonal form would allow this great improvement in speed to be realized.

In section 2 we describe recent work on tridiagonalization algorithms, particularly the early work of Wachspress [13] and the later work of Geist [3], which forms the basic algorithm on which our recovery methods are built. In section 3 we discuss methods for detecting when breakdown will occur, and in section 4 we describe new recovery algorithms that considerably broaden the class of matrices that can be reduced to tridiagonal form with elementary similarity transformations. In section 5 we apply these recovery algorithms to matrices that are known to be difficult to reduce. Section 6 contains our conclusions.

2. Recent Studies

Dax and Kaniel [2] described experiments with reduction from upper Hessenberg form to tridiagonal form using elementary similarity transformations. During the reduction, they monitored the size of the multipliers as follows. They defined a *control parameter* for the reduction of row k as $m_k = max_{i>k+1}|H_{k,i}/H_{k,k+1}|$. If m_k was greater than a specified value, then *breakdown* was said to have occurred, and their algorithm aborted. They observed that for 100 random test matrices of order 50×50 the number of breakdowns as a function of the specified value m was:

$m = 2^r$	r =	16	12	10	8	7
breakdowns		0	1	5	20	41

Dax and Kaniel referred to Wilkinson's detailed error analysis in [16] and concluded that there was a low probability of having large errors in eigenvalues computed with the tridiagonal matrix, even when using control parameters as large as 2^{16}. They report that their results differed from EISPACK by as much as 10^{-7} for their test matrices.

Watkins [15] made Dax and Kaniel's reduction from Hessenberg to tridiagonal form more robust by incorporating a recovery method, performing an implicit shifted LR iteration on the matrix when breakdown occurs. The iteration preserves the structure of the partially reduced matrix and produces a matrix similar to the pre-recovery matrix. Moreover, it often eliminates the breakdown condition. The cost of the LR iterations can be significant, but the iteration brings the matrix closer to triangular form, so the work is not entirely wasted even if breakdown persists.

Wachspress' reduction algorithm is applied after the matrix is first reduced to upper Hessenberg form by Householder transformations. The reduction is performed by columns, starting with the last column. If a large multiplier is encountered, the offending column is skipped rather than reduced. This yields a "striped" tridiagonal matrix with a few columns containing nonzeros above the superdiagonal as shown:

$$\begin{pmatrix} \times & \times & & \times & & \times & \\ \times & \times & \times & \times & & \times & \\ & \times & \times & \times & & \times & \\ & & \times & \times & \times & \times & \\ & & & \times & \times & \times & \\ & & & & \times & \times & \times \\ & & & & & \times & \times \end{pmatrix}$$

The stripes have little effect on the arithmetic complexity of the ADI iteration, since nonzeros do not propagate from the stripes during the solution of the ADI equations.

When Wachspress developed a precursor to the recovery method described in section 4, the earlier scheme of leaving a striped matrix was abandoned. Instead, a reduction algorithm was used that attempts to eliminate all columns of a Hessenberg matrix one at a time starting with column n using elementary transformations. He applied his recovery method whenever multipliers larger than 1000 were detected.

Hare and Tang [5] describe a reduction method where the unitary transformations used to reduce the matrix to Hessenberg form H and the elementary transformations used to reduce H to tridiagonal form are interlaced. The interlacing allows a permutation that reduces the number of multipliers potentially greater than 1 from $O(n^2)$ to $O(n)$. Hare and Tang also employ a recovery method if any of these multipliers exceed a preset bound. They observe that the relative error of the eigenvalues grows as the bound is increased, and at their recommended setting of 100, they observed errors as large as 10^{-8}.

In [3], Geist presents an algorithm that reduces the original matrix directly to tridiagonal form, avoiding the intermediate Hessenberg form. The algorithm reduces column 1, then row 1, and so on down the matrix using a method of threshold pivoting to control the size of the multipliers. Elementary similarity transformations are used throughout the reduction. In his report, the multiplier bound is set to 10, and a variation of Wachspress' recovery method is employed whenever this bound is exceeded.

Geist reported that this bound was not exceeded when reducing random dense matrices, and the accuracy of the eigenvalues did not show the degradation observed by Dax and Kaniel [2]. His report also presented a class of matrices that are difficult to reduce to tridiagonal form. Throughout this paper we will refer to Geist's algorithm as the *threshold pivoting* approach.

3. Anatomy of Breakdown

Because the threshold pivoting approach appears the most successful of recent tridiagonalization methods, we have chosen to study how this method breaks down.

At step k of the threshold pivoting algorithm, the matrix has the form shown in Figure 1. If v or $w^T = 0$, then the matrix has been deflated, and step k can be skipped.

$$\left(\begin{array}{cc|cc} S_{k-1} & & & \\ & & \times & \\ \hline & \times & \alpha & w^T \\ & & v & B_{k+1} \end{array}\right)$$

Figure 1: Partially reduced matrix.

If v and $w^T \neq 0$, then the threshold pivoting approach first permutes rows and columns greater than k to maximize the product $|v_1 w_1|$. Let m_k be the maximum multiplier at step k. If $m_k < m_{max}$, where m_{max} is a prescribed value chosen to control error growth, column k and row k are reduced by elementary similarity transformations. An inherent measure of the stability of this reduction is the value of $|cos(v, w)|$, as shown below.

Theorem 1. *Using Figure 1 as a reference, let m_c and m_r be the maximum multipliers in column k and row k respectively. If column k is reduced first, then*

$$\begin{aligned} m_c &= \frac{max_{i>1}|v_i|}{|v_1|} \\ m_r &= \frac{|v_1| max_{i>1}|w_i|}{\|v\| \ \|w\| \ |cos(v,w)|}. \end{aligned}$$

Proof. Given that column k is reduced first by an elementary similarity transformation

RAR^{-1}, the expression for m_c is obvious. The form of R is

$$\left(\begin{array}{c|c} I_{k-1} & \\ \hline & G_k \end{array}\right)$$

and after the transformation is applied, $\bar{v} = Gv$ and $\bar{w}^T = w^T G^{-1}$. Thus, $\bar{w}^T \bar{v} = w^T G^{-1} G v = w^T v$. And since only $\bar{v}_1 \neq 0$, $\bar{w}_1 \bar{v}_1 = w^T v$. Since R is elementary, $\bar{v}_1 = v_1$ so $v_1 \bar{w}_1 = w^T v$ or $\bar{w}_1 = \frac{w^T v}{v_1}$. Therefore,

$$m_r = \frac{max|w_i|}{|\bar{w}_1|} = \frac{|v_1| \; max|w_i|}{|w^T v|} \quad ; \quad i > 1.$$

Substituting $|w^T v| = \| v \| \; \| w \| \; |cos(v, w)|$ we get the result.

Corollary 1. *An upper bound on m_r is given by $m_r \leq \frac{1}{|cos(v,w)|}$ and is independent of the method used to reduce column k.*

Proof $\frac{\bar{v}_1}{\|v\|} \leq 1$ and $\frac{max|w|}{\|w\|} \leq 1$.

Several observations are possible given these results. Reducing v with a unitary transformation will maximize $\bar{v}_1$, and since $\bar{w}_1 \bar{v}_1 = w^T v$, $\bar{w}_1$, the pivot for the reduction of row k, is minimized. For this reason we do not recommend using unitary transformations during the reduction to tridiagonal form.

In the threshold pivoting approach, if $w^T v = 0$, then $max(|v_i|, |w_i|)$ is permuted into the pivot position before the recovery routine is begun. This permutation attempts to minimize the maximum product $m_c m_r$. This effect can be seen by forming the product of the terms m_c and m_r in Theorem 1.

There can be occasions when the $max(m_c, m_r)$ is smaller when row k is reduced before column k. This happens when for some $i > 1$

$$\begin{array}{ccc} |w_1| max|v_i| & > & |v_1| max|w_i| \\ & and & \\ \frac{max|w_i|}{|w_1|} & < & \frac{max|v_i|}{|v_1|}. \end{array}$$

But the threshold pivoting algorithm as given in [3] always reduces the column first.

One goal of threshold pivoting could be to minimize the maximum multiplier over the entire reduction. This goal is not practical because of the high complexity required.

But it is possible to find the pivot that minimizes the maximum multiplier at each step k efficiently using the expressions for m_c, m_r, and their row-first counterparts just discussed. It requires only $n-k$ flops to evaluate each of the four expressions for all possible pivot choices plus an additional $n-k$ flops to evaluate the inner product. We have implemented this pivot selection scheme and compared it to the original heuristic pivoting scheme, which chooses the permutation that maximizes $|v_1 w_1|$. First, the additional time required to evaluate m_c and m_r is insignificant compared to the overall reduction time, as expected. Second, we found this pivot scheme, which can be viewed as a greedy algorithm, is not as good as the heuristic scheme at minimizing the maximum multiplier over the entire reduction. For random matrices larger than 80×80, the optimal method often failed to meet the prescribed tolerance. Moreover, the eigenvalues were at least as accurate and often better using the original heuristic pivoting scheme.

An alternate goal of threshold pivoting is to minimize the condition number of M, where MAM^{-1} is tridiagonal. Let $M = R_{n-2}C_{n-2}R_{n-3}C_{n-3} \cdots R_1C_1$, where C_k reduces column k, and R_k reduces row k. A suggested heuristic [6] that tends to reduce this condition number is to minimize the maximum entry in R_kC_k at each step. The structure of R_kC_k is

$$\left(\begin{array}{c|cc} I_{k-1} & & \\ \hline & \alpha & r^T \\ & c & I_{k+1} \end{array}\right)$$

where c_i are the column multipliers, r_i are the row multipliers, and $\alpha = 1 + r^Tc$. (If the row is reduced first, then C_kR_k has a dense $(n-k) \times (n-k)$ submatrix, and the minimization is very difficult.) We have already described how the multipliers can be minimized. The α term can be simplified to $\alpha = 2 - v_1w_1/|w^Tv|$ (using the notation in Figure 1), which allows the efficient evaluation of α for all possible permutations at step k. Notice that the original heuristic term v_1w_1 appears in α and may explain why this heuristic often works well. We have implemented a pivoting scheme that minimizes the maximum entry in R_kC_k and found it to be comparable to the original pivoting scheme in accuracy. The new pivoting scheme is also more robust than the original heuristic because it leads to a smaller maximum multiplier over the entire reduction. We are continuing to investigate other heuristics, which also attempt to minimize the condition number of M. The original threshold pivoting scheme can be improved by using the new heuristic scheme.

4. Recovery from Breakdown

In most of the tridiagonalization schemes, including the threshold pivoting scheme, breakdown is defined to be the situation where a multiplier has exceeded some tolerance. When breakdown occurs, a number of options are available to circumvent the problem. Sometimes a local transformation can decrease the size of the multiplier such that the reduction can continue [5], but local methods cannot be robust because the tridiagonal form is unique once the first column and row of the transformation matrix M are fixed [8]. Thus, if the unique form has a small pivot, breakdown cannot be avoided without changing the first row or column. In [16], Wilkinson states that if a breakdown occurs, one can go back to the beginning and apply the transformation $N_1 A N_1{}^{-1}$ in the hope breakdown will not occur again. No method of choosing N_1 has been found that guarantees that the breakdown condition found in A will not exist in $N_1 A N_1{}^{-1}$. This choice is still an open research area.

The initial recovery method proposed by Wachspress [13] is to return to the beginning (in his case, the bottom) of the matrix and apply an N_1 of the form

$$N_1 = \left(\begin{array}{c|cc} I & \multicolumn{2}{c}{O} \\ \hline O & 1 & p \\ & 0 & 1 \end{array}\right)$$

where p is initially set to 1. This transformation changes the three nonzero entries in row n and column n and introduces a nonzero in the $(n-2, n)$ position. This "bulge" is then chased up the matrix to the point of breakdown with elementary similarity transformations. The procedure is mathematically equivalent to an implicit shift LR iteration that is truncated at the point of breakdown. Assuming the breakdown occurs at column k, this chasing procedure fills in column $k+1$, which must then be annihilated to return the matrix to its pre-recovery structure. The elimination of column $k+1$ requires $O(k^2)$ flops and accounts for the majority of the work performed during the recovery. If breakdown occurs during the recovery, or if the original breakdown condition persists after the recovery step, the recovery method is repeated with

p incremented. After a fixed number of consecutive recovery failures, the algorithm aborts.

A modification of this recovery method is described in Geist [3] for the threshold pivoting approach. There are four main differences. First, since the threshold pivoting approach reduces the matrix starting with row 1 and column 1, rather than column n, the bulge is formed at the top of the matrix and chased down to the breakdown. Second, since the breakdown can occur in the column or row reduction, two forms of N_1 are used to start the recovery depending on whether the large multiplier is above or below the diagonal. They are shown in Figure 2. Third, the value p used in Wachspress' recovery

$$\left(\begin{array}{cc|c} 1 & \times & \multirow{2}{*}{O} \\ 0 & 1 & \\ \hline \multicolumn{2}{c|}{O} & I \end{array}\right) \qquad \left(\begin{array}{cc|c} 1 & 0 & \multirow{2}{*}{O} \\ \times & 1 & \\ \hline \multicolumn{2}{c|}{O} & I \end{array}\right)$$

Figure 2: Two Forms of N_1 used in threshold pivoting recovery.

method is replaced with a random value uniformly distributed on [.1,1]. Finally, after three consecutive recovery failures, the multiplier tolerance is increased by a factor of 10. Then if the tolerance ever exceeds 1000, the algorithm aborts.

We propose two further modifications to the existing threshold pivoting recovery method. These modifications are designed to help avoid the problems reported in [3] that follow from Theorem 2.

Theorem 2. *Using Figure 1 as a reference, let b be the subdiagonal part of the first column of B_{k+1}, and let u^T be w^T without element w_1. Assume the pivot $w_1 = 0$. If $u^T b = 0$ then the threshold pivoting recovery method will not change the value of w_1.*

Proof. Assume the recovery method is applied, and a bulge is being chased down the matrix. When the bulge is eliminated at $k-2$, row $k-1$ becomes βu^T, where β is the multiplier formed at step $k-2$. The small pivot, w_1, is unaffected by the recovery up to this point. Let c be the element $(k-1, k)$. After row $k-1$ is eliminated a second time, $\bar{w_1} = w_1 + \frac{\beta}{c} u^T b$.

The first modification addresses the problem when $u^T b = 0$. This modification has a natural column version dual to the row procedure, just as the bulge chase does, so we will describe only the row version. Assume the situation described in Theorem 2. Let

$|b_i|$ be the maximum value in the first column of B_{k+1}. Add row i to row k and subtract column k from column i, so that w_1 is now the largest element in column $k+1$. This local similarity transformation also introduces a nonzero in position $(k-1, i)$. Now apply $N_1 A N_1{}^{-1}$ and chase the bulge. The nonzero is eliminated, along with the rest of row $k-1$, during the bulge chase. The local transformation not only increases the magnitude of w_1, it also changes the value of $w^T v$. Since N_1 is random, $|w^T v|$ cannot be guaranteed to increase, although if the value is small, there is a high probability that it will.

There are still cases when this modified recovery method will fail. The example shown in Figure 3 is such a case [3]. A small number of consecutive failures of the

$$A = \begin{pmatrix} 20 & 1 & & & & & \\ & 19 & 1 & & & & \\ & & 18 & 1 & & & \\ & & & \cdot & \cdot & & \\ & & & & \cdot & \cdot & \\ & & & & & \cdot & 1 \\ 1 & & & & & & 1 \end{pmatrix}$$

Figure 3: A matrix for which the modified recovery method fails.

recovery method is usually indicative of a matrix with a large number of small inner products. Rather than abort after repeated failure of the modified recovery method, the following method will often find a similar matrix that does not have this property. The algorithm is simple and efficient to apply, requiring only $O(n^2)$ flops to execute. It involves applying $N_2 N_1 A_0 N_1{}^{-1} N_2{}^{-1}$ where A_0 is the original matrix and N_1 and N_2 have the form shown in Figure 4. Column 1 of N_1 and row 1 of N_2 contain random numbers uniformly distributed on [0,1]. It should be considered a method of last resort because it requires restarting the reduction.

$$\begin{pmatrix} \alpha & 0 \\ v & I \end{pmatrix} \quad \begin{pmatrix} \gamma & u^T \\ 0 & I \end{pmatrix}$$

Figure 4: Forms of N_1 and N_2 matrices.

5. Results

We incorporated these recovery methods into the threshold pivoting algorithm and compared their performance against the original reduction algorithms.

We present the results of three test matrices that exercise each of the features of the modified algorithm. The first matrix is representative of a class of Lyapunov matrices. Our interest in this particular 48×48 dense matrix arose when Wachspress' algorithm failed to reduce it to tridiagonal form because of off-tridiagonal growth. His attempts to circumvent this problem by permuting rows and columns prior to the Hessenberg reduction also failed because the Hessenberg matrix did not preserve the desired scaling. In contrast, the threshold pivoting algorithm reduces this same matrix to tridiagonal form with no multipliers exceeding 10. The success of the threshold pivoting approach on this matrix was due to the algorithm's ability to move large off-tridiagonal entries to the superdiagonal or subdiagonal and thus avoid the large off-tridiagonal growth.

The second matrix came from a set of unsymmetric eigenvalue test matrices found in [4] and has the form

$$\begin{pmatrix} -1 & -1 & -1 & -1 & -1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}.$$

Repeated breakdown was observed for an 8×8 version of this matrix. When Wachspress' algorithm was used, over 30 recovery steps were required to reduce the matrix to tridiagonal form, and multipliers as large as 2961 were observed. When the original threshold pivoting algorithm was used, 5 consecutive recovery steps were required. After the 5 recovery steps filled in the matrix at step 2 of the reduction, the algorithm proceeded to the end without further problems. For larger versions of this matrix both of the above methods fail because they exceed their bounds on consecutive recovery steps.

When the improved threshold pivoting algorithm was applied to the 8×8 matrix, one recovery was performed at step 2 and one was performed at step 4 of the reduction. The incorporation of the local similarity transformation into the recovery routine eliminated the need for consecutive recoveries in this problem. Larger versions of this matrix can be reduced using the improved algorithm.

The third test matrix is shown in Figure 3 and is described in [3] as a matrix the original threshold pivoting algorithm failed to reduce. The original algorithm aborted after 9 consecutive recovery attempts before step 4 of the reduction. Off-tridiagonal growth occurs rapidly for this matrix, and its structure prevents threshold pivoting from permuting all the large elements to the superdiagonal and subdiagonal. The incorporation of the local similarity transformation had little effect on this behavior. However, rather than abort, the new algorithm applied the second recovery method discussed in section 4. The algorithm then reduced the modified matrix to tridiagonal form without further breakdowns.

The accuracy of the modified threshold pivoting algorithm is often better than that of the original algorithm. The eigenvalues of the tridiagonal matrix differ from the eigenvalues of the original matrix by as much as 10^{-14} for the three test matrices. We do not see an increase in the number of breakdowns as the matrix size grows as reported in [5], but we do observe a decrease in accuracy as the matrix size increases. For example, reducing dense random matrices of order less than 20 produced errors of 10^{-15}, but matrices of order 200 produced errors as large as 10^{-10}. Since multipliers routinely exceed 1 in our reduction, the correlation between error and matrix size is not unexpected.

6. Conclusions

We have analyzed the conditions under which reduction to tridiagonal form breaks down or becomes unstable, and have shown how this behavior is characterized by $|cos(v, w)|$, where v is the column being reduced and w^T is the corresponding row. We have discussed how heuristic methods of threshold pivoting attempt to minimize the maximum multipliers used during the reduction, and how unitary transformations can maximize the maximum multipliers.

We have presented efficient methods for recovering from breakdown when it occurs. These recovery methods require only $O(n^2)$ flops and significantly extend the class of matrices that can be reduced to tridiagonal form. One of the methods requires the reduction to restart at column 1 and row 1 and is recommended only as a method of last resort. Moreover, complete failure of tridiagonalization is not catastrophic. One can always resort to QR reduction for those relatively few cases where the faster but less robust method fails.

The Lyapunov equation solver with ADI iteration applied to the tridiagonal system succeeded as predicted by theory with no difficulties once the matrix was reduced to tridiagonal form. The modified threshold pivoting method is now being added to the solver to improve the robustness of the reduction step. The incorporation into the solver will require the accumulation of the transformations, which could also be used to calculate the eigenvectors of the matrix. A detailed report on the numerical solution of the Lyapunov matrix equation is in preparation.

7. References

[1] R. H. Bartels and G. W. Stewart. A solution of the matrix equation $AX + XB = C$. *Comm. ACM*, 15:820 – 826, 1972.

[2] A. Dax and S. Kaniel. The ELR method for computing the eigenvalues of a general matrix. *SIAM J. Numer. Anal.*, 18(4):597 – 605, August 1981.

[3] G. A. Geist. *Reduction of a general matrix to tridiagonal form.* Technical Report, Oak Ridge National Laboratory, February 1989. ORNL/TM-10991.

[4] R. T. Gregory and D. L. Karney. *A Collection of Matrices for Testing Computational Algorithms.* Robert E. Krieger Publishing Company, 1978.

[5] D. E. Hare and W. P. Tang. *Toward a stable tridiagonalization algorithm for general matrices.* Technical Report, University of Waterloo, January 1989. CS-89-03.

[6] Linda Kaufman. Personal Communication.

[7] C. D. LaBudde. The reduction of an arbitrary real square matrix to tridiagonal form using similarity transformations. *Math. Comp.*, 17:443 – 447, 1963.

[8] B. N. Parlett. Personal Communication.

[9] B. N. Parlett. A note on LaBudde's algorithm. *Math. Comp.*, 19:505 – 506, 1964.

[10] B. T. Smith, J. M. Boyle, J. J. Dongarra, B. S. Garabow, Y. Ikebe, V. C. Klema, and C. B. Moler. *Matrix Eigensystem Routines - EISPACK Guide.* Springer, Heidelberg, 1974.

[11] C. Strachey and J. G. F. Francis. The reduction of a matrix to codiagonal form by elimination. *The Computer Journal*, 4:168, 1961.

[12] E. L. Wachspress. The ADI minimax problem for complex spectra. In *Iterative Methods for Large Linear Systems*, Academic Press, 1989. To Appear.

[13] E. L. Wachspress. ADI solution of Lyapunov equations. October 1988. Talk at MSI Workshop on Practical Iterative Methods for Large-scale Computations, Minneapolis, MN.

[14] E. L. Wachspress. Iterative solution of the Lyapunov matrix equation. *Applied Mathematics Letters*, 1:87 – 90, 1987.

[15] D. Watkins. Use of the LR algorithm to tridiagonalize a general matrix. July 1988. Talk at SIAM Annual Meeting, Minneapolis, MN.

[16] J. H. Wilkinson. *The Algebraic Eigenvalue Problem.* Oxford University Press, Oxford, 1965.

ORNL/TM-11089

INTERNAL DISTRIBUTION

1. B. R. Appleton
2. J. B. Drake
3-7. G. A. Geist
8-9. R. F. Harbison
10. M. T. Heath
11. M. R. Leuze
12-16. F. C. Maienschein
17. E. G. Ng
18. G. Ostrouchov
19. B. W. Peyton
20-24 S. Raby
25. C. H. Romine
26-30. R. C. Ward
31. P. H. Worley
32. A. Zucker
33. J. J. Dorning (Consultant)
34. R. M. Haralick (Consultant)
35. Central Research Library
36. ORNL Patent Office
37. K-25 Plant Library
38. Y-12 Technical Library Document Reference Station
39. Laboratory Records - RC
40-41. Laboratory Records Department

EXTERNAL DISTRIBUTION

42. Dr. Donald M. Austin, Office of Scientific Computing, Office of Energy Research, ER-7, Germantown Building, U.S. Department of Energy, Washington, DC 20545

43. Dr. Zhaojun Bai, Courant Institute of Mathematical Sciences, 251 Mercer St., New York, NY 10012

44. Dr. Jesse L. Barlow, Department of Computer Science, Pennsylvania State University, University Park, PA 16802

45. Dr. Chris Bischof, Mathematics and Computer Science Division, Argonne National Laboratory, 9700 South Cass Avenue, Argonne, IL 60439

46. Prof. Ake Bjorck, Department of Mathematics, Linkoping University, Linkoping 58183, Sweden

47. Dr. Daniel Boley, Department of Computer Science, University of Minnesota, 200 Union St. S.E. Rm.4-192 Minneapolis, MN 55455

48. Dr. James C. Browne, Department of Computer Sciences, University of Texas, Austin, TX 78712

49. Dr. Bill L. Buzbee, Scientific Computing Division, National Center for Atmospheric Research, P.O. Box 3000, Boulder, CO 80307

50. Dr. Ralph Byers , Department of Mathematics, University of Kansas, Lawrence, KA 66045

51. Dr. Donald A. Calahan, Department of Electrical and Computer Engineering, University of Michigan, Ann Arbor, MI 48109

52. Dr. Tony Chan, Department of Mathematics, University of California, Los Angeles, 405 Hilgard Avenue, Los Angeles, CA 90024

53. Dr. Jagdish Chandra, Army Research Office, P.O. Box 12211, Research Triangle Park, North Carolina 27709

54. Dr. Eleanor Chu, Department of Computer Science, University of Waterloo, Waterloo, Ontario, Canada N2L 3G1

55. Dr. Mel Ciment, National Science Foundation, 1800 G Street, NW, Washington, DC 20550

56. Prof. Tom Coleman, Department of Computer Science, Cornell University, Ithaca, NY 14853

57. Dr. Paul Concus, Mathematics and Computing, Lawrence Berkeley Laboratory, Berkeley, CA 94720

58. Dr. Jane K. Cullum, IBM T. J. Watson Research Center, P.O. Box 218, Yorktown Heights, NY 10598

59. Dr. George Cybenko, Computer Science Department, University of Illinois, Urbana, IL 61801

60. Dr. George J. Davis, Department of Mathematics, Georgia State University, Atlanta, GA 30303

61. Dr. James Demmel, Courant Institute of Mathematical Sciences, 251 Mercer St., New York, NY 10012

62. Dr. Jack J. Dongarra, Mathematics and Computer Science Division, Argonne National Laboratory, 9700 South Cass Avenue, Argonne, IL 60439

63. Dr. Iain Duff, CSS Division, Harwell Laboratory, Didcot, Oxon OX11 ORA, England

64. Prof. Pat Eberlein, Department of Computer Science, SUNY/Buffalo, Buffalo, NY 14260

65. Dr. Stanley Eisenstat, Department of Computer Science, Yale University, P.O. Box 2158 Yale Station, New Haven, CT 06520

66. Dr. Lars Elden, Department of Mathematics, Linkoping University, 581 83 Linkoping, Sweden

67. Dr. Howard C. Elman, Computer Science Department, University of Maryland, College Park, MD 20742

68. Dr. Albert M. Erisman, Boeing Computer Services, 565 Andover Park West, Tukwila, WA 98188

69. Dr. Geoffrey C. Fox, Booth Computing Center 158-79, California Institute of Technology, Pasadena, CA 91125

70. Dr. Paul O. Frederickson, LNASA Ames Research Center, RIACS, M/S 230-5, Moffett Field, CA 94035

71. Dr. Fred N. Fritsch, L-300, Mathematics and Statistics Division, Lawrence Livermore National Laboratory, P.O. Box 808, Livermore, CA 94550

72. Dr. Robert E. Funderlic, Department of Computer Science, North Carolina State University, Raleigh, NC 27650

73. Dr. Dennis B. Gannon, Computer Science Department, Indiana University, Bloomington, IN 47405

74. Dr. David M. Gay, Bell Laboratories, 600 Mountain Avenue, Murray Hill, NJ 07974

75. Dr. C. William Gear, Computer Science Department, University of Illinois, Urbana, Illinois 61801

76. Dr. Alan George, Vice President, Academic and Provost, Needles Hall, University of Waterloo, Waterloo, Ontario, Canada N2L 3G1

77. Dr. John Gilbert, Xerox Palo Alto Research Center, 3333 Coyote Hill Road, Palo Alto, CA 94304

78. Prof. Gene H. Golub, Department of Computer Science, Stanford University, Stanford, CA 94305

79. Dr. Joseph F. Grcar, Division 8331, Sandia National Laboratories, Livermore, CA 94550

80. Dr. Per Christian Hansen, UCI*C Lyngby, Building 305, Technical University of Denmark, DK-2800 Lyngby, Denmark

81. Dr. Don E. Heller, Physics and Computer Science Department, Shell Development Co., P.O. Box 481, Houston, TX 77001

82. Dr. Nicholas Higham, Department of Computer Science, Cornell University, Ithaca, NY 14853-7501

83. Dr. Charles J. Holland, Air Force Office of Scientific Research, Building 410, Bolling Air Force Base, Washington, DC 20332

84. Dr. Robert E. Huddleston, Computation Department, Lawrence Livermore National Laboratory, P.O. Box 808, Livermore, CA 94550

85. Dr. Ilse Ipsen, Department of Computer Science, Yale University, P.O. Box 2158 Yale Station, New Haven, CT 06520

86. Ms. Elizabeth Jessup, Department of Computer Science, Yale University, P.O. Box 2158, Yale Station, New Haven, CT 06520

87. Dr. Harry Jordan, Department of Electrical and Computer Engineering, University of Colorado, Boulder, CO 80309

88. Dr. Bo Kagstrom, Institute of Information Processing, University of Umea, 5-901 87 Umea, Sweden

89. Dr. Hans Kaper, Mathematics and Computer Science Division, Argonne National Laboratory, 9700 South Cass Avenue, Argonne, IL 60439

90. Dr. Linda Kaufman, Bell Laboratories, 600 Mountain Avenue, Murray Hill, NJ 07974

91. Dr. Robert J. Kee, Applied Mathematics Division 8331, Sandia National Laboratories, Livermore, CA 94550

92. Ms. Virginia Klema, Statistics Center, E40-131, MIT, Cambridge, MA 02139

93. Dr. Alan J. Laub, Department of Electrical and Computer Engineering, University of California, Santa Barbara, CA 93106

94. Dr. Robert L. Launer, Army Research Office, P.O. Box 12211, Research Triangle Park, North Carolina 27709

95. Dr. Charles Lawson, Applied Mathematics Group, Jet Propulsion Laboratory, California Institute of Technology, M/S 506-232, 4800 Oak Grove Drive, Pasadena, CA 91109

96. Prof. Peter D. Lax, Director, Courant Institute of Mathematical Sciences, New York University, 251 Mercer Street, New York, NY 10012

97. Dr. John G. Lewis, Boeing Computer Services, P.O. Box 24346, M/S 7L-21, Seattle, WA 98124-0346

98. Dr. Heather M. Liddell, Director, Center for Parallel Computing, Department of Computer Science and Statistics, Queen Mary College, University of London, Mile End Road, London E1 4NS, England

99. Dr. Joseph Liu, Department of Computer Science, York University, 4700 Keele Street, Downsview, Ontario, Canada M3J 1P3

100. Dr. Franklin Luk, Electrical Engineering Department, Cornell University, Ithaca, NY 14853

101. Dr. Thomas A. Manteuffel, Computing Division, Los Alamos National Laboratory, Los Alamos, NM 87545

102. Dr. Paul C. Messina, California Institute of Technology, Mail Code 158-79, Pasadena, CA 91125

103. Dr. Cleve Moler, Ardent Computers, 550 Del Ray Avenue, Sunnyvale, CA 94086

104. Dr. Dianne P. O'Leary, Computer Science Department, University of Maryland, College Park, MD 20742

105. Maj. C. E. Oliver, Office of the Chief Scientist, Air Force Weapons Laboratory, Kirtland Air Force Base, Albuquerque, NM 87115

106. Dr. James M. Ortega, Department of Applied Mathematics, University of Virginia, Charlottesville, VA 22903

107. Prof. Chris Paige, Department of Computer Science, McGill University, 805 Sherbrooke Street W., Montreal, Quebec, Canada H3A 2K6

108. Prof. Roy P. Pargas, Department of Computer Science, Clemson University, Clemson, SC 29634-1906

109. Prof. Beresford N. Parlett, Department of Mathematics, University of California, Berkeley, CA 94720

110. Prof. Merrell Patrick, Department of Computer Science, Duke University, Durham, NC 27706

111. Dr. Robert J. Plemmons, Departments of Mathematics and Computer Science, North Carolina State University, Raleigh, NC 27650

112. Dr. Alex Pothen, Department of Computer Science, Pennsylvania State University, University Park, PA 16802

113. Dr. John K. Reid, CSS Division, Building 8.9, AERE Harwell, Didcot, Oxon, England OX11 0RA

114. Dr. Noah Rhee, Department of Mathematics, University of Missouri-Kansas City, Kansas City, MO 64110-2499

115. Dr. John R. Rice, Computer Science Department, Purdue University, West Lafayette, IN 47907

116. Dr. Garry Rodrigue, Numerical Mathematics Group, Lawrence Livermore Laboratory, Livermore, CA 94550

117. Dr. Donald J. Rose, Department of Computer Science, Duke University, Durham, NC 27706

118. Dr. Ahmed H. Sameh, Computer Science Department, University of Illinois, Urbana, IL 61801

119. Dr. Michael Saunders, Systems Optimization Laboratory, Operations Research Department, Stanford University, Stanford, CA 94305

120. Dr. Robert Schreiber, RIACS, Mail Stop 230-5, NASA Ames Research Center, Moffet Field, CA 94035

121. Dr. Martin H. Schultz, Department of Computer Science, Yale University, P.O. Box 2158 Yale Station, New Haven, CT 06520

122. Dr. David S. Scott, Intel Scientific Computers, 15201 N.W. Greenbrier Parkway, Beaverton, OR 97006

123. Dr. Lawrence F. Shampine, Mathematics Department, Southern Methodist University, Dallas, TX 75275

124. Dr. Kermit Sigmon, Department of Mathematics, University of Florida, Gainesville, FL 32611

125. Dr. Danny C. Sorensen, Mathematics and Computer Science Division, Argonne National Laboratory, 9700 South Cass Avenue, Argonne, IL 60439

126. Prof. G. W. Stewart, Computer Science Department, University of Maryland, College Park, MD 20742

127. Dr. Daniel B. Szyld, Department of Computer Science, Duke University, Durham, NC 27706-2591

128. Dr. W.-P. Tang, Department of Computer Science, University of Waterloo, Waterloo, Ontario, Canada N2L 3G1

129. Prof. Charles Van Loan, Department of Computer Science, Cornell University, Ithaca, NY 14853

130. Prof. James M. Varah, The University of British Columbia, 2053-2324 Main Mall, Vancouver, B.C., Canada V6T 1W5

131. Dr. Robert G. Voigt, ICASE, MS 132-C, NASA Langley Research Center, Hampton, VA 23665

132. Dr. E. L. Wachspress, Department of Mathematics, University of Tennessee, Knoxville, TN 37996-1300

133. Dr. D. S. Watkins, Department of Pure and Applied Mathematics, Washington State University, Pullman, WA 99164-2930

134. Dr. Andrew B. White, Computing Division, Los Alamos National Laboratory, Los Alamos, NM 87545

135. Dr. Margaret Wright, Bell Laboratories, 600 Mountain Avenue, Murray Hill, NJ 07974

136. Dr. Marvin Wunderlich National Security Agency, Ft. George G. Meade, MD 20755

137. Office of Assistant Manager for Energy Research and Development, U.S. Department of Energy, Oak Ridge Operations Office, P.O. Box 2001 Oak Ridge, TN 37831-8600

138-147. Office of Scientific & Technical Information, P.O. Box 62, Oak Ridge, TN 37831