ornl

OAK RIDGE
NATIONAL
LABORATORY

MARTIN MARIETTA

ORNL/TM-10991

Reduction of a General Matrix to Tridiagonal Form

G. A. Geist

OAK RIDGE NATIONAL LABORATORY
CENTRAL RESEARCH LIBRARY
CIRCULATION SECTION
4500N ROOM 175
LIBRARY LOAN COPY
DO NOT TRANSFER TO ANOTHER PERSON
If you wish someone else to see this report, send in name with report and the library will arrange a loan.

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

Printed in the United States of America. Available from
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Road, Springfield, Virginia 22161
NTIS price codes—Printed Copy: A03; Microfiche A01

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

ORNL/TM-10991

Engineering Physics and Mathematics Division

Mathematical Sciences Section

REDUCTION OF A GENERAL MATRIX TO TRIDIAGONAL FORM

G. A. Geist

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

Date Published - March 1989

Research was supported by the Applied Mathematical Sciences Research Program of the Office of Energy Research, U.S. Department of Energy.

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

MARTIN MARIETTA ENERGY SYSTEMS LIBRARIES

3 4456 0290332 0

Contents

1 Introduction 1

2 Recent Approaches 2

3 A Threshold Pivoting Approach 7

4 Results 9

5 Observations 11

6 Summary 13

7 References 14

REDUCTION OF A GENERAL MATRIX TO TRIDIAGONAL FORM

G. A. Geist

Abstract

This report reviews recent research in finding eigenvalues of general matrices by first reducing the matrix to tridiagonal form and then applying LR iteration to this reduced form. The report also describes a new algorithm for solving this problem and shows why it is superior to the other approaches. While our results show the algorithm to be over three times faster than the EISPACK routines for finding eigenvalues, there are some matrices for which the algorithm fails. The report describes how to detect these matrices during the reduction process and how to attempt to proceed.

1. Introduction

The standard method for computing all of the eigenvalues of a dense matrix is based on the QR iteration scheme [3]. In this scheme, orthogonal (QR) similarity transformations are successively applied to the matrix to reduce it to triangular form, so that the eigenvalues appear on the diagonal. Repeated application of these transformations to a general matrix is prohibitively expensive, however, so that in practice the original matrix is first reduced to a simpler form that can be preserved during the subsequent iterative phase. For a general matrix, the initial reduction is usually to upper Hessenberg form (triangular except for one additional sub-diagonal) by elementary or orthogonal similarity transformations. The initial reduction to Hessenberg form requires $O(n^3)$ operations, where n is the order of the matrix. Computation of the eigenvalues of the reduced matrix usually requires only a few QR iterations per eigenvalue, totaling another $O(n^3)$ operations. Both the initial and iterative phases are costly, but less than the cost of iterating directly with the original matrix. This two-phase approach is implemented in the standard EISPACK software for the general eigenvalue problem [9].

If the original matrix is symmetric, then that symmetry can easily be preserved in the initial reduction, so that the result is in fact tridiagonal. Although the reduction to tridiagonal form costs $O(n^3)$ operations, the subsequent iterations preserve the tridiagonal form and are much less expensive, so that the total cost of the iterative phase is reduced to $O(n^2)$ operations. Again, standard software is available in EISPACK implementing this two-phase approach for the symmetric case [9].

The attractively low operation count of iterating with a tridiagonal matrix suggests that the tridiagonal form would be extremely beneficial in the nonsymmetric case as well. There are two difficulties with such an approach: First, QR iteration does not preserve the structure of a nonsymmetric tridiagonal matrix. This problem can be overcome by using LR iteration [8] instead, which preserves the tridiagonal form. Second, it is difficult to reduce a nonsymmetric matrix to tridiagonal form by similarity transformations in a numerically stable manner. This second problem is the primary focus of this paper.

In the early 1960's there was a great amount of interest and research devoted to finding a stable way to reduce a general matrix via similarity transformations to

tridiagonal form. Very little of this research appeared in the literature because no satisfactory solution was found to the problem [2]. The problem is addressed in some detail by Wilkinson [13], and several algorithms are given. But the overall conclusion in [13] was that no general purpose practical algorithm existed. Because of his conclusions little research was directed at the problem of reduction to tridiagonal form for nearly 15 years.

In 1981 Dax and Kaniel published a paper [1] that has inspired renewed interest in the problem of reducing a general nonsymmetric matrix to tridiagonal form in order to compute the eigenvalues of the original matrix. In the next section we review the work of Dax and Kaniel, as well as subsequent work, focusing on the various attempts to maintain numerical stability during the tridiagonal reduction of a general matrix. We describe our new approach in section 3 and give some experimental results in section 4. In section 5 we make some observations on the inherent numerical difficulty of the problem and identify a class of matrices that tax the ability of any algorithm. The final section contains our conclusions.

2. Recent Approaches

Dax and Kaniel presented an algorithm for computing all the eigenvalues of a full nonsymmetric matrix in [1]. Their approach was to reduce A to upper Hessenberg form H using stabilized elementary similarity transformations [13]. They then proceeded to zero out each row i of H from $H_{i,i+2}$ to $H_{i,n}$ using $H_{i,i+1}$ as the pivot element in an elementary similarity transformation. Unlike the initial reduction to H, where pivoting is performed to stabilize the transformations, this stage of the reduction cannot use pivoting without destroying Hessenberg form. Moreover, if one attempts to recover the Hessenberg form by eliminating the nonzeros that were introduced, the inverse of the previous similarity transformation is required, which returns the matrix entries to their pre-pivot values.

The difficulty that may be encountered using this scheme is illustrated by the matrix H, shown in Figure 1. A straightforward attempt to proceed by reducing row 3 would entail a division by zero. The inability to pivot can cause numerical difficulties even if $H_{i,i+1}$ is not zero. The ratios of $H_{i,i+2}/H_{i,i+1}, \ldots, H_{i,n}/H_{i,i+1}$ are called the elimination factors or multipliers. Because they have the potential to magnify errors, it is desirable

$$\begin{pmatrix} \times & \times & & & & \\ \times & \times & \times & & & \\ & \times & \times & 0 & \times & \times \\ & & \times & \times & \times & \times \\ & & & \times & \times & \times \\ & & & & \times & \times \end{pmatrix}$$

Figure 1: Breakdown during reduction to tridiagonal form.

for these multipliers to be small (less than one). If any of these multipliers is much larger than one, then the potential for amplifying numerical errors increases. During the reduction to Hessenberg form, the multipliers are guaranteed to be less than one because pivoting is performed. But in the second step, where rows are eliminated, the multipliers can be arbitrarily large.

Dax and Kaniel addressed the problem of large multipliers by monitoring their size during the reduction. If the multipliers ever exceeded a user-defined value, then the algorithm aborted, and the user was forced to retreat to Hessenberg form and apply QR iteration. The bound on the multipliers chosen for the majority of their test runs was 2^{16}. Given this tolerance, they bounded the error term and concluded that in most modern computers, enough digits are represented in the mantissa that the possibility of exceeding the tolerance is quite rare in practice.

If H is successfully reduced to tridiagonal form T, then the next step of their algorithm is to apply a version of LR iteration designed specifically for nonsymmetric tridiagonal matrices. The algorithm applies either single-shift or double-shift iterations depending on the value of the shifts. In addition to incorporating most of the ideas given in [13], Dax and Kaniel also incorporated some new variations in arbitrary shifts when LR breaks down. For instance, if an eigenvalue is not found after 50 iterations, then the algorithm returns to the original T and applies the present shift to it.

Dax and Kaniel observed that for their test problems their algorithm required only 17% of the time required by their implementation of the EISPACK routines. They also observed that the frequency of breakdowns increased with the size of the matrix, requiring more arbitrary shifts.

Tang considered the problem of reducing A to tridiagonal form and presented an

algorithm in [10]. This algorithm is similar to a method described in [13] for avoiding breakdown during the reduction phase. Wilkinson did not give an algorithm because this method for avoiding breakdown did not work after the first half of the matrix had been reduced. Tang's algorithm first reduces A to H using orthogonal similarity transformations. Unless H can be reduced directly to T without the multipliers growing much larger than one, Tang's algorithm will reduce A to a "comrade" matrix, which is tridiagonal except that the last column may be dense (see Figure 2).

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

Figure 2: Form of a comrade matrix.

No empirical results are given in his technical report, but the author claims the method is stable. It is unclear, however, whether an efficient algorithm exists for finding the eigenvalues of a general comrade matrix. More recently, Hare and Tang have begun investigating the effects of interleaving orthogonal and elementary similarity transformations during the reduction [7].

Two other researchers have described different approaches to this problem. They were all influenced by [1], and they all focused on the fact that Dax and Kaniel had ignored possible ways to recover from breakdown during the reduction to T. Wilkinson states on page 404 of [13] that

> If breakdown occurs in the rth step of the reduction of a Hessenberg matrix to tri-diagonal form we must return to the beginning and compute $N_1 A N_1^{-1}$ for some N_1 in the hope that failure will be avoided in this matrix.

This recovery method is actually too restrictive. In the approaches that follow, the key feature will be efficient methods for finding matrices similar to A without returning to the beginning and wasting work already performed on the matrix.

Wachspress became interested in this problem while trying to solve the Lyapunov matrix equation: $AX + XA^T = C$ [11]. His desire was to reduce A to T and then solve

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

Figure 3: Reduction of Hessenberg matrix from the bottom.

$TZ + ZT^T = C'$ by an alternating direction implicit (ADI) method. His approach to reducing A to T involves first reducing A to H by Householder transformations. Then, unlike previous approaches, he begins the reduction of H from the bottom, first eliminating column n then $n-1$, etc. (See Figure 3).

In Wachspress' approach [5], the sizes of the multipliers are monitored just as they are in Dax and Kaniel's method, but the maximum value allowed is much lower, on the order of 2^{10}. When this value is exceeded, Wachspress applies what we will term a 2×2 *fixup*. Let H^* be the partially reduced Hessenberg matrix. The 2×2 *fixup* involves applying the similarity transformation BH^*B^{-1}, where B has the structure shown in Figure 4.

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

Figure 4: Form of Matrix B used to start 2×2 *fixup*.

The transformation introduces a single nonzero at $H_{n-2,n}$. This "bulge" is then "chased" up the second super-diagonal to the point where the original breakdown occurred. While conceptually one can think of the bulge as simply replacing the small number that caused the breakdown, in reality a very different process occurs.

Assume that H^* has encountered a very small number, ϵ, on the super-diagonal in column k, and the bulge from the 2×2 *fixup* has been chased up to $H^*_{k,k+2}$, as shown in Figure 5a. At the next step of the recovery, column $k+1$ fills in (Figure 5b). It is actually the process of eliminating column $k+1$ with a similarity transformation

that produces a larger value in the $k, k+1$ position. At this point we have a matrix with the same structure as the original H^* but with a larger value at $k, k+1$. Thus the reduction to tridiagonal form can proceed as long as the new value in $H^*_{k-1,k}$ is sufficiently large. If not, the 2×2 *fixup* is repeated using a different value for α. If an unfortunate choice for α causes the non-pivoting bulge-chase to fail, the fixup is restarted with another α.

$$\begin{pmatrix} \times & \times & \times & & & \\ \times & \times & \epsilon & & & \\ & \times & \times & \times & \times & \\ & & \times & \times & \times & \\ & & & \times & \times & \times \\ & & & & \times & \times \end{pmatrix} \qquad \begin{pmatrix} \times & \times & \times & + & & \\ \times & \times & \epsilon & + & & \\ & \times & \times & \times & & \\ & & \times & \times & \times & \\ & & & \times & \times & \times \\ & & & & \times & \times \end{pmatrix}$$

$a.$ $\qquad\qquad$ $b.$

Figure 5: Matrix during 2×2 *fixup*.

The advantage of Wachspress' recovery method is the small operation count required to find a similar matrix with the same structure as the one for which breakdown occurred. To generate and chase the bulge up to position $H_{k,k+2}$ requires only $5(n-k)$ flops. The rest of the recovery involves eliminating column $k+1$ a second time, which has a cost of $O(k^2)$. Conceptually, the reduction from H to T proceeds until a breakdown occurs, then the recovery backs up one column, and the reduction continues.

The name 2×2 *fixup* for the recovery step comes from the size of the block in the lower right corner of B and can be extended to blocks of size 3×3 and larger in the obvious way. Since the reduction is unique once the first column is fixed, this family of fixups can be viewed as modifying the first column in such a way that the breakdown condition disappears. While arbitrary changes to a few entries in the first column appear to work very well in practice, more research needs to be done to determine which entries should be changed and by what amount to ensure that the reduction can continue.

Watkins has also worked on finding the eigenvalues of nonsymmetric matrices by reduction to tridiagonal form [12]. Like Wachspress, his approach first reduces A to H with Householder transformations. But unlike Wachspress, the reduction from H to T proceeds from the top by rows. The reduction proceeds as in Dax and Kaniel's method

as long as the multipliers are less than 2^{16}. If the multipliers exceed this value, Watkins proposes the following recovery method. Apply one implicit shift LR iteration to H^* and try to proceed with the reduction to tridiagonal form. The LR iteration produces a matrix similar to the original, so it meets the main criterion of a recovery method. Since the form of H^* is preserved by the LR iteration, only the nonzero entries in H^* are modified in his implementation of the LR algorithm. The iterations are repeated until the reduction can proceed or until all the eigenvalues of A are found.

This recovery method is more expensive than Wachspress' method, but it has the advantage that if the iteration fails to find a similar matrix that can be reduced to tridiagonal form, it still has been useful for finding the eigenvalues. It is possible with Watkins' method to find some eigenvalues during the recovery process, deflate the matrix, and then continue with the reduction. This allows the reduction to tridiagonal form to be much more robust than in Dax and Kaniel's approach.

If the reduction to T succeeds, then single-shift and double-shift LR iterations are applied to T in a manner similar to Dax and Kaniel's algorithm but without some of their arbitrary shift conditions. Watkins' approach during this stage is closer to the ideas in Wilkinson [13]. As might be expected, Watkins' results for finding the eigenvalues of T were similar to Dax and Kaniel's, both in terms of speed and accuracy. (Both noted that the computation of the eigenvalues was less accurate with their approaches than with the EISPACK approach.)

3. A Threshold Pivoting Approach

Our approach differs from all the other approaches described so far in that we do not reduce A to H first. Instead, we attempt to reduce rows and columns of A simultaneously with elementary similarity transformations. The algorithm begins with row and column 1 of A and proceeds down the matrix. At any point in the reduction the matrix has the form shown in Figure 6 with a dense lower block E and a tridiagonal upper block.

At each step of the reduction, a variant of threshold pivoting is performed in E, in an attempt to minimize the multipliers in both the row and column being eliminated. In [1], the multipliers used in the column reduction to H were all less than one, but the multipliers used in the row reduction to T could be arbitrarily large. Our approach

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

Figure 6: Structure of the matrix during our reduction.

tries to balance this process. Instead of choosing the pivot that guarantees all the column multipliers will be less than one, we choose a pivot that will allow some column multipliers to be greater than one if it causes the row multipliers to be reasonably small. The magnitude of both the row and column multipliers is required to be less than a user-defined tolerance. For our tests this tolerance was set to 10. In general, we would like to minimize the maximum multiplier, but we have been unable to discover an efficient scheme to find those pivot elements. Our pivot selection is heuristic, but seems to work quite well in practice. We choose the pivot index that maximizes the product of the corresponding row and column elements.

Regardless of the pivot selection, there are still simple examples where the maximum product is zero. In this case some form of recovery must be applied. We chose to implement a variation of the 2×2 *fixup*. In our implementation, B has one of the forms shown in Figure 7.

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

Figure 7: Two Forms of Matrix B used to start our 2×2 *fixup*.

The choice of B is determined by whether the large multipliers are above or below the diagonal. There are two reasons for not using an LR iteration as in Watkins [12]. First, the 2×2 *fixup* is computationally equivalent to some implicit single shift LR iteration that is truncated at the point of breakdown. Thus the 2×2 *fixup* is similar to one iteration of LR, but always less costly. Second, since E is dense, we have no

means of determining a "good" shift for an LR iteration.

Assume the maximum product is zero and the matrix does not deflate, i.e., the maximum of the row or column is not zero. (If the matrix deflates, then the reduction can skip immediately to the next row and column.) Our approach chooses the pivot index of the maximum element of the row and column, performs this pivot, and then applies a 2×2 *fixup*. If three applications of 2×2 *fixups* fail to find a similar matrix that allows the reduction to proceed, then the tolerance is automatically increased (with suitable warning to the user) and the 2×2 *fixups* continue until the tolerance reaches 10^5. At this point the algorithm aborts. This has been observed only in specially contrived matrices, which we describe in more detail in section 5.

Given that A has been successfully reduced to T, we then apply implicit double shift LR to T, with simple arbitrary shifts if needed. Unlike other approaches described in this report, we do not mix single and double shift approaches. This decision was based on two facts. First, using only the implicit double shift algorithm allowed a cleaner implementation of the algorithm to find the eigenvalues of T. Second, since this algorithm's complexity is only $O(n^2)$, it is a low order effect in the overall process and thus a cleaner implementation was felt to be more important than reducing the constant in front of a low order term.

Wilkinson suggested performing a diagonal similarity transformation to T before applying LR to it. All the researchers who applied LR to T did apply this transformation. Given a nonsymmetric tridiagonal matrix, Wilkinson noted that this matrix could be transformed such that its superdiagonal is all unity. Moreover, this superdiagonal is invariant under LR iterations. Thus all algorithms for finding the eigenvalues of T described here operate only on the diagonal, which is converging to the eigenvalues, and the subdiagonal, which is converging to zero. It is interesting to note that finding the eigenvalues of a nonsymmetric tridiagonal matrix requires the same amount of storage as the symmetric case when using LR. Similarly, the nonsymmetric case has the same number of operations per iteration as the symmetric case using LR.

4. Results

Figure 8 illustrates the advantage of our method over Dax and Kaniel's method. The example is from page 89 of Gregory and Karney [6]. While both methods produced a

tridiagonal matrix that is similar to the original matrix, it is clear that an LR iteration applied to Dax and Kaniel's T is much more likely to encounter problems than LR applied to our $\bar{T}$, because of the wide variation of scale in their T. In particular, notice that the elimination factor in column 2 would be 10^4. The values in their matrix range from 10^{-4} to 10^8, while ours range from 0.3 to 3.2.

$$A = \begin{pmatrix} .4163 & .3176 & 0 & 0 & 0 \\ .0001 & .4132 & .8175 & 0 & 0 \\ .6321 & .3157 & .4823 & .6614 & 0 \\ .5157 & .8321 & .5642 & .6541 & .4321 \\ .5563 & .4431 & .2567 & .8325 & .8475 \end{pmatrix}$$

$$T = \begin{pmatrix} .4163 & .3176 & 0 & 0 & 0 \\ .0001 & 5167. & .8175 & 0 & 0 \\ 0 & -32659398. & -5166. & .6614 & 0 \\ 0 & 0 & -.0001 & .5615 & .4321 \\ 0 & 0 & 0 & .5463 & .4006 \end{pmatrix}$$

$$\bar{T} = \begin{pmatrix} .1809 & .3176 & 0 & 0 & 0 \\ -.1721 & -1.243 & .8175 & 0 & 0 \\ 0 & -3.287 & 2.359 & .6614 & 0 \\ 0 & 0 & .4391 & .8685 & .4321 \\ 0 & 0 & 0 & .3338 & .6469 \end{pmatrix}$$

Figure 8: Comparison of matrices produced by Dax and Kaniel's method (T) and our method ($\bar{T}$).

Table 1 compares our algorithm with the comparable EISPACK routines: ELMHES, ORTHES, and HQR for a series of test matrices ranging in size from 50 to 300. The matrices were random with the entries distributed uniformly over the interval [0,1]. ELMHES reduces A to H using stabilized elementary similarity transformations while ORTHES reduces A to H using Householder transformations. The ORTHES results are given because ORTHES is the first step in several of the methods presented in section 2. HQR finds the eigenvalues of H using an implicit double shift QR iteration. Our algorithms are presented in the table as A2TRI and TLR. A2TRI reduces A directly to T as described in section 3. TLR finds the eigenvalues of T. The error in the eigenvalues from TLR and HQR were comparable in these tests. For these matrices the multipliers never exceeded 10 in our algorithm, and we feel this may be the reason

we did not see the degradation in accuracy that Dax and Kaniel and Watkins observed. All computations were performed in double precision on a SUN 3/280.

	EISPACK			NEW	
n	ELMHES	ORTHES	HQR	A2TRI	TLR
50	2.70	4.66	12.92	4.00	0.64
100	21.58	37.60	92.08	32.84	2.56
128	44.92	80.06	208.62	70.34	4.32
150	72.28	136.88	334.56	112.58	5.42
200	173.20	314.78	667.78	266.36	9.86
250	338.54	631.42	1388.36	510.52	14.14
300	582.64	1136.34	2305.14	994.58	28.98

Table 1: Execution times in seconds for our new routines and the standard EISPACK routines.

It is clear from the table that our method can find the eigenvalues of a dense nonsymmetric matrix much faster than the EISPACK routines. A complexity analysis, where low order terms are ignored, shows that TLR requires $5n$ flops per iteration versus $4n^2$ flops for HQR. While the number of iterations varies between TLR and HQR, they both require only a few, often less than 5, iterations per eigenvalue. Further, the arithmetic complexity of ELMHES is $\frac{5}{6}n^3$ while the complexity of A2TRI is $\frac{3}{2}n^3$ flops, assuming A2TRI needs to apply the 2×2 *fixup* only a constant number of times. The results in Table 1 reflect speedups greater than 3 for A2TRI/TLR over ELMHES/HQR. This is consistent with the relative complexities of the routines.

5. Observations

While our algorithm A2TRI extends the class of matrices that can be reduced to tridiagonal form, there are still matrices for which the recovery methods presented cannot succeed. A practical code must be able to detect when such a matrix is encountered and react accordingly. This section describes a class of such matrices and reveals why reducing A to H by Householder transformations, as done by Wachspress and Watkins, actually aggravates the problem of reducing H to T.

Wachspress [5] made the following observation, which is the key to defining a class

of matrices that are difficult to reduce to tridiagonal form. Computations of the form:

$$\begin{pmatrix} I_{k-1} & 0 \\ 0 & G_{n-k} \end{pmatrix} \begin{pmatrix} T_{k-1} & & \\ & \alpha & w^T \\ & v & B \end{pmatrix} \begin{pmatrix} I_{k-1} & 0 \\ 0 & G_{n-k}^{-1} \end{pmatrix} = \begin{pmatrix} T_{k-1} & & \\ & \alpha & w^T G^{-1} \\ & Gv & GBG^{-1} \end{pmatrix}$$

preserve the inner product of the kth row and column, since $w^T G^{-1} G v = w^T v$.

This property affects any algorithm that reduces A to tridiagonal form by similarity transformations, not just the ones described in this report. Consider the following example. Let the partially reduced matrix have the form

$$\begin{pmatrix} T & & \\ & \alpha & w^T \\ & v & B \end{pmatrix}$$

and let $w^T v = 0$. After all but the first entry of v have been eliminated, the matrix has the form

$$\begin{pmatrix} T & & & \\ & \alpha & 0 & \bar{w}^T \\ & \beta & B' & \\ & 0 & & \end{pmatrix}$$

The first entry in w^T is zero and it must be used as the pivot. This is true regardless of the pivot selection in v or whether orthogonal transformations are used to eliminate v.

The above observation explains why methods that reduce A to H using Householder transformations have more breakdowns. Since the orthogonal transformation preserves $\| v \|_2$, it maximizes the one entry left in v. The product of this entry and the eventual pivot on the other side of the diagonal is constant; therefore, the pivot element is being minimized. This in turn leads to large multipliers and breakdown.

If $w^T v = 0$, then a breakdown condition will occur no matter what transformation is used, but a 2×2 *fixup* will allow the reduction to continue provided that $\bar{w}^T B_{*,1} \neq 0$, where $B_{*,1}$ is the first column of B. If this quantity is zero, then a 3×3 *fixup* is required to continue. Larger fixups are required as more consecutive dot products of w^T and the

columns of B equal zero. The probability of two or more consecutive zero dot products occurring is extremely small in practice. Moreover, it is a simple matter to check the dot products before beginning the recovery step and adjust the recovery if necessary. For these reasons, our algorithm appears to be a practical alternative to HQR in most cases.

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

Figure 9: A matrix which A2TRI cannot reduce to tridiagonal form.

We can contrive matrices that are extremely difficult to reduce with any method. Figure 9 shows a simple, well conditioned matrix that cannot be reduced to tridiagonal form with A2TRI. Notice that at the very first step the dot product between row 1 and column 1 is zero, and the dot product between row 1 and every other subdiagonal column is also zero. However, A2TRI can find a tridiagonal matrix similar to the one shown in Figure 9 by reducing NAN^{-1} for a suitability chosen N.

6. Summary

We have presented an algorithm for reducing nonsymmetric matrices to similar tridiagonal matrices (A2TRI) and an algorithm for finding the eigenvalues of the tridiagonal matrices (TLR). A2TRI incorporates threshold pivoting and a 2×2 *fixup* method, which makes it one of the most robust algorithms available. A complexity analysis reveals that our algorithms are over three times faster than the conventional method of finding the eigenvalues of a dense nonsymmetric matrix, i.e. the ELMHES and HQR algorithms of EISPACK. Our empirical results show A2TRI requires about twice as much time as ELMHES, which is not unreasonable since it eliminates twice as many entries in the original matrix. The algorithm TLR runs in time proportional to n^2, while HQR runs in time proportional to n^3. Illustrating this difference, a problem that

required 2305 seconds using HQR, finished in only 29 seconds with TLR.

We have described a class of matrices that are difficult to reduce to tridiagonal form regardless of the method used, and we show how these matrices can be detected by computing dot products of particular rows and columns. This can be done when a breakdown occurs during the reduction to determine which recovery method to use. We plan to incorporate this feature into A2TRI in the near future.

We are developing a parallel version of the A2TRI algorithm. The parallel algorithms that have been implemented for finding the eigenvalues of nonsymmetric matrices are inefficient for large numbers of processors [4]. Because of our radically different approach, our new algorithms should not suffer from the same bottlenecks. Preliminary empirical results appear to confirm this conjecture. Details of the parallel implementation and results will be given in a future report.

7. References

[1] 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.

[2] P. L. Eberlein and J. M. Ortega. Personal Communications.

[3] J. G. F. Francis. The QR transformation - Part 2. *The Computer Journal*, 4:332–345, 1961.

[4] G. A. Geist and G. J. Davis. *Finding eigenvalues and eigenvectors of unsymmetric matrices using a distributed-memory multiprocessor.* Technical Report, Oak Ridge National Laboratory, November 1988. ORNL/TM-10938.

[5] G. A. Geist, A. Lu, and E. L. Wachspress. *Stabilized Gaussian reduction of an arbitrary matrix from Hessenberg to tridiagonal form.* Technical Report, Oak Ridge National Laboratory, February 1989. ORNL/TM-11089.

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

[7] 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.

[8] H. Rutishauser. Solution of eigenvalue problems with the LR transformation. *Nat. Bur. Standards Appl. Math. Ser.*, 49:47 – 81, 1958.

[9] 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.

[10] W. P. Tang. *A stabilized algorithm for tridiagonalization of an unsymmetric matrix.* Technical Report, University of Waterloo, Waterloo, Ontario, April 1988. CS-88-14.

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

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

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

INTERNAL DISTRIBUTION

1. B. R. Appleton
2. J. B. Drake
3-7. G. A. Geist
8-9. R. F. Harbison
10. M. T. Heath
11-15. J. K. Ingersoll
16. M. R. Leuze
17-21. F. C. Maienschein
22. E. G. Ng
23. G. Ostrouchov
24. B. W. Peyton
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. Lawrence J. Baker, Exxon Production Research Company, P.O. Box 2189, Houston, TX 77252-2189

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

97. 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

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

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

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

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

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

103. Dr. Brent Morris, National Security Agency, Ft. George G. Meade, MD 20755

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

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

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

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

134. Dr. Arthur Wouk, Army Research Office, P.O. Box 12211, Research Triangle Park, North Carolina 27709

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

136. 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

137-146. Office of Scientific & Technical Information, P.O. Box 62, Oak Ridge, TN 37831