A Case Against a Divide and Conquer Approach to the Nonsymmetric Eigenvalue Problem

E.R. Jessup

CU-CS-554-91 November 1991

A Case Against a Divide and Conquer Approach to the Nonsymmetric Eigenvalue Problem

E.R. Jessup
Department of Computer Science
University of Colorado, Boulder 80309–0430

CU-CS-554-91 November 1991

A Case Against a Divide and Conquer Approach to the Nonsymmetric Eigenvalue Problem

E.R. Jessup *
Department of Computer Science
University of Colorado
Boulder, Colorado 80309

November 1991

Abstract

Divide and conquer techniques based on rank-one updating have proven fast, accurate, and efficient in parallel for the real symmetric tridiagonal and unitary eigenvalue problems and for the bidiagonal singular value problem. Although the divide and conquer mechanism can also be adapted to the real nonsymmetric eigenproblem in a straightforward way, most of the desirable characteristics of the other algorithms are lost. In this paper, we examine the problems of accuracy and efficiency that can stand in the way of a nonsymmetric divide and conquer eigensolver based on low-rank updating.

1 Introduction

The eigenvalues and eigenvectors of a real nonsymmetric matrix A have traditionally been computed by first reducing A to Hessenberg form H and then computing the eigendecomposition of H by the QR method. The serial nature of the QR method combined with the high cost of data transfer on distributed-memory multiprocessors has made parallel implementations of this approach inefficient [16]. The failure of parallel QR algorithms has sparked recent research into new algorithms including stabilized tridiagonalization [9, 14], iterative refinement techniques [11], and homotopy methods [25].

In this paper, we examine the use of divide and conquer techniques based on low-rank updating for solving the real nonsymmetric eigenvalue problem. This divide and conquer approach was first applied to the symmetric tridiagonal eigenvalue problem by Cuppen [8] and analyzed and implemented in parallel by Dongarra and Sorensen [12]. The algorithm involves tearing a symmetric tridiagonal matrix T into a pair of symmetric tridiagonal submatrices T_1 and T_2 by removing off-diagonal elements of T and using rank-one updating techniques to form the eigendecomposition of T from those of T_1 and T_2. We review the method in Section 2 of this paper.

The symmetric divide and conquer technique of [8, 12] provides a fast and accurate serial alternative to the QR method or to bisection with inverse iteration [12, 23]. In addition, the

*This research was supported in part by the Applied Mathematical Sciences Research Program, Office of Energy Research, U.S. Department of Energy, under contract DE-AC05-84OR21400 with Martin Marietta Energy Systems, Inc. and in part by National Science Foundation grant CCR-9109785.

divide and conquer method is efficient when implemented on shared-memory multiprocessors [12]. Similar methods have also performed well for the bidiagonal singular value problem [24] and for the unitary eigenvalue problem [4]. In this paper, we show that the efficiency and accuracy shared by these divide and conquer methods cannot be expected in general when the approach is applied to the nonsymmetric eigenproblem.

In Sections 3 and 4 of this paper, we demonstrate how the symmetric divide and conquer method of [8, 12] can be extended in a straightforward way to nonsymmetric tridiagonal and Hessenberg eigenproblems if we assume that the submatrices formed from tearing are diagonalizable. This algorithm allows us to examine the troubles with a nonsymmetric divide and conquer eigensolver in direct analogy to the symmetric eigensolver. In [1], Adams and Arbenz consider a general rank-r update to a nonsymmetric matrix without assuming diagonalizability. Their proposed algorithm bears less resemblance to the symmetric algorithm, but many of the conclusions that we will draw about our simplified algorithm do apply to their complete theory. In fact, we will show that one major obstacle to accurate implementation of such a method arises for virtually any updating method that does not employ the original matrix T at some stage after tearing. We discuss the difficulties that can plague a nonsymmetric divide and conquer method in Section 5. We present our conclusions in Section 6.

Throughout this paper, unless otherwise specified, capital Greek and Roman letters represent matrices, lower case Roman letters represent column vectors, and lower case Greek letters represent scalars. A superscript T denotes transpose, a superscript H denotes conjugate transpose, and $\bar{h}$ is the element-wise complex conjugate of the vector h. The vector e_j is the j-th "canonical vector" with all elements equal to zero except the j-th which equals 1.

2 A Review of the Symmetric Tridiagonal Method

An unreduced symmetric tridiagonal matrix T of order $n = 2m$ can be written as the matrix sum

$$T = \begin{pmatrix} T_1 & \\ & T_2 \end{pmatrix} + \theta\beta \begin{pmatrix} e_m \\ \theta^{-1} e_1 \end{pmatrix} \begin{pmatrix} e_m^T & \theta^{-1} e_1^T \end{pmatrix}, \tag{1}$$

where β is the mth off-diagonal element of T, e_i is the ith unit vector of length m, and T_1 and T_2 are symmetric tridiagonal of order m. The algorithm can be made backward stable with $\theta = \text{sign}(e_m^T T_1 e_m)$ [5].

If the solutions to the two smaller eigensystems are $T_1 = X_1 D_1 X_1^T$ and $T_2 = X_2 D_2 X_2^T$, then

$$T = X \left[D + \beta\theta \begin{pmatrix} l_1 \\ \theta^{-1} f_2 \end{pmatrix} \begin{pmatrix} l_1^T & \theta^{-1} f_2^T \end{pmatrix} \right] X^T$$

where

$$X = \begin{pmatrix} X_1 & \\ & X_2 \end{pmatrix}, \quad D = \begin{pmatrix} D_1 & \\ & D_2 \end{pmatrix},$$

$l_1^T = e_m^T X_1$ is the last row of X_1, and $f_2^T = e_1^T X_2$ is the first row of X_2. To solve the eigenproblem for T, it is necessary to find the eigenvalues and eigenvectors of the diagonal plus rank-one matrix

$$D + \rho z z^T = X^T T X,$$

where $z^T = \sqrt{\frac{\beta\theta}{\rho}} \begin{pmatrix} l_1^T & \theta^{-1} f_2^T \end{pmatrix}$, and ρ is selected so that $\| z \|_2 = 1$ [8].

The eigensystem of T is computed via the rank-one updating technique described in [6, 18]. Namely, if all elements of z are non-zero and if the diagonal elements of D are distinct, then the eigenvalues of $D + \rho zz^T$ are the roots $\lambda_1 > \ldots > \lambda_n$ of the *secular equation*

$$w(\lambda) = 1 + \rho z^T (D - \lambda)^{-1} z = 1 + \sum_{j=1}^{n} \frac{(e_j^T z)^2}{\delta_j - \lambda}.$$

If $\beta > 0$ and the diagonal elements of D are given by $\delta_1 > \ldots > \delta_n$, each eigenvalue is bracketed by the adjacent diagonal elements of D: $\delta_i > \lambda_i > \delta_{i+1}$ and $\delta_1 + \rho z^T z > \lambda_1 > \delta_1$. When $\beta < 0$, a change of variables leads to a similar result. This interlacing property means that the roots of $w(\lambda)$ may be found efficiently using any one-dimensional root finder such as the one based on rational interpolation developed in [6]. Once λ_j has been found, its corresponding eigenvector u_j is computed from

$$u_j = \frac{(D - \lambda_j)^{-1} z}{\| (D - \lambda_j)^{-1} z \|_2}.$$

When the diagonal elements of D are not distinct, *i.e.*, $\delta_l = \delta_{l+1} = \ldots = \delta_{l+k}$, the eigenproblem of order n is reduced to one of order $n - k$ by the process of *deflation*. The eigenvector basis is first rotated to zero out the elements $\zeta_{l+1}, \ldots, \zeta_{l+k}$ corresponding to the multiple elements $\delta_{l+1} = \cdots = \delta_{l+k}$: a product of plane rotations G_l is applied so that

$$G_l(\zeta_l, \zeta_{l+1}, \cdots, \zeta_{l+k})^T = (\zeta_l', \zeta_{l+1}', \cdots, \zeta_{l+k}')^T = (\zeta_l', 0, \cdots, 0)^T.$$

For $l + 1 \le j \le l + k$, the jth eigenvalue in exact arithmetic is the jth element of D ($\lambda_j = \delta_j$), and its corresponding eigenvector is the appropriate canonical vector ($u_j = e_j$) [6].

Representing the product of all rotations by the matrix G, the matrix T is expressed as $T = XG^T Q \Lambda Q^T G X^T = U \Lambda U^T$, where $U \Lambda U^T$ is the eigendecomposition of $G(D + \rho zz^T)G^T$. The eigenvalues of T are the diagonal elements of Λ, and the eigenvectors of T are the columns of $U = XG^T Q$.

The above derivation assumes exact arithmetic. Deflation rules have also been developed for finite precision in [12]: rotations are applied when diagonal elements of D are close, and deflation occurs when elements of z are small. Numerical experiments have confirmed that the increase in speed due to this deflation is substantial for serial and shared-memory parallel implementations [12]. With appropriate choice of deflation criteria and use of extended precision, it is possible to guarantee computation of highly accurate eigenvalues and orthogonal eigenvectors [30].

Implementations of the divide and conquer method [12, 22] recursively subdivide the symmetric tridiagonal matrix T until the resulting subproblems are of a desired order. In parallel implementations, the smallest subproblems are solved in parallel with one problem per processor, and the work to solve larger order subproblems is shared by more than one processor [12, 22]. In particular, high parallel efficiency has been achieved on shared-memory multiprocessors by dynamically assigning independent root-finding and eigenvector computing tasks to separate processors at each level of updating [12]. A parallel implementation of the divide and conquer method can also be pipelined with reduction of a symmetric matrix to tridiagonal form [10, 12].

3 A Nonsymmetric Eigensolver

In this section, we investigate the application of low-rank updating techniques to the nonsymmetric tridiagonal eigenproblem. This problem arises as a subproblem in other numerical methods such as exponential interpolation [2, 3]. Tridiagonal eigenproblems also result from nonsymmetric Lanczos algorithms [19] or the stabilized reduction of general eigenproblems to tridiagonal form [9, 15]. We show in Section 4 that the theory we develop in this section for the tridiagonal problem extends in a straightforward manner to Hessenberg matrices. We assume the use of exact arithmetic in both cases. The nonsymmetric algorithms presented in this paper require that the submatrices be formed by the matrix tearing be diagonalizable.

The method of Section 2 was based on rank-one updating of a torn symmetric tridiagonal matrix. The theory behind that method does not carry through to solving a nonsymmetric updating problem $D + uv^T$. Furthermore, the updating matrix uv^T is dense and otherwise unstructured. In this paper, we employ a rank-two tearing of a nonsymmetric matrix that leads to smaller order subproblems, a shorter recursion tree, and a convenient arrowhead updating matrix. A symmetric tridiagonal eigensolver based on rank-two updating appears to be competitive with the symmetric eigensolver based on rank-one updating [13]. Solution of symmetric arrowhead eigenproblems is discussed in [28, 31].

Let T be the tridiagonal matrix with diagonal elements $\alpha_1, \ldots, \alpha_n$, sub-diagonal elements $\gamma_1, \ldots, \gamma_{n-1}$, and super-diagonal elements $\beta_1, \ldots, \beta_{n-1}$, and suppose $n = 2m + 1$. By splitting off two superdiagonal elements β_m and β_{m+1} and the corresponding subdiagonal elements γ_m and γ_{m+1}, we can write the matrix T in terms of the tridiagonal submatrices T_1 and T_2:

$$T = \left(\begin{array}{c|c|c} T_1 & & \\ \hline & \alpha_m & \\ \hline & & T_2 \end{array}\right) + \left(\begin{array}{c|c|c} & \beta_m & \\ \hline \gamma_m & & \beta_{m+1} \\ \hline & \gamma_{m+1} & \end{array}\right).$$

If T_1 and T_2 are diagonalizable, we can compute the eigendecompositions $T_1 = X_1 D_1 X_1^{-1}$ and $T_2 = X_2 D_2 X_2^{-1}$ with diagonal matrices D_1 and D_2. Substituting these decompositions and abbreviating $\alpha = \alpha_m$ gives the matrix product

$$\begin{aligned} T &= \left(\begin{array}{c|c|c} X_1 D_1 X_1^{-1} & & \\ \hline & \alpha & \\ \hline & & X_2 D_2 X_2^{-1} \end{array}\right) + \left(\begin{array}{c|c|c} & \beta_m & \\ \hline \gamma_m & & \beta_{m+1} \\ \hline & \gamma_{m+1} & \end{array}\right) \\ &= \hat{X} \left[\left(\begin{array}{c|c|c} D_1 & & \\ \hline & \alpha & \\ \hline & & D_2 \end{array}\right) + \left(\begin{array}{c|c|c} & v_1 & \\ \hline h_1^T & & h_2^T \\ \hline & v_2 & \end{array}\right) \right] \hat{X}^{-1}, \end{aligned} \qquad (2)$$

where

$$\hat{X} = \left(\begin{array}{c|c|c} X_1 & & \\ \hline & 1 & \\ \hline & & X_2 \end{array}\right),$$

$v_1 = \beta_m X_1^{-1} e_m$, $v_2 = \gamma_{m+1} X_2^{-1} e_1$, $h_1 = \gamma_m X_1^T e_m$, and $h_2 = \beta_{m+1} X_2^T e_1$ for canonical vectors e_1 and e_m of appropriate length.

We permute the elements of equation (2) to form

$$T = X\left[\left(\begin{array}{cc|c} D_1 & & \\ & D_2 & \\ \hline & & \alpha \end{array}\right) + \left(\begin{array}{cc|c} & & v_1 \\ & & v_2 \\ \hline h_1^T & h_2^T & 0 \end{array}\right)\right] X^{-1} \tag{3}$$

with

$$X = \left(\begin{array}{cc|c} X_1 & & \\ & X_2 & \\ \hline & & 1 \end{array}\right)$$

and rewrite the interior matrix of equation (3) as

$$M = \left(\begin{array}{cc|c} D_1 & & \\ & D_2 & \\ \hline & & \alpha \end{array}\right) + \left(\begin{array}{cc|c} & & v_1 \\ & & v_2 \\ \hline h_1^T & h_2^T & 0 \end{array}\right) = \begin{pmatrix} D & v \\ h^T & \alpha \end{pmatrix}.$$

The arrowhead matrix M is the sum of a diagonal matrix and a rank two nonsymmetric matrix. The eigenvalues of the matrix M are the eigenvalues of T. The left and right eigenvectors of M premultiplied by X^{-H} and X, respectively, are the left and right eigenvectors of T.

The procedure for computing the remaining eigenvalues and eigenvectors of M follows basic steps similar to those for the eigendecomposition of a diagonal plus symmetric rank one matrix developed in [6, 18], but because M is nonsymmetric, the details differ in several important ways.

If M has diagonal elements $\delta_1, \ldots, \delta_{n-1}$, α, last column $(v_1, \ldots, v_{n-1}, \alpha)^T$, and last row $(h_1, \ldots, h_{n-1}, \alpha)$, then the following lemmas establish when diagonal elements of M can be retained as eigenvalues of T.

Lemma 3.1 *If the diagonal elements $\delta_1, \ldots, \delta_{n-1}$ of M are distinct, the element δ_j is an eigenvalue of M if and only if $v_j h_j = 0$.*

Lemma 3.2 *If M has repeated diagonal elements $\delta_i = \delta_j$ with $h_i h_j \neq 0$, M is similar to a matrix with $\delta_i = \delta_j$ and $h_i = 0$.*

Proof: The proof is by construction of the unitary similarity transformation that reduces h_i to zero when $\delta_i = \delta_j$. Let

$$\tau^2 = |h_j|^2 + |h_i|^2, \quad c = \frac{h_j}{\tau}, \quad s = \frac{h_i}{\tau},$$

then the matrix is transformed in the following way

$$\begin{pmatrix} \bar{c} & -\bar{s} & 0 \\ s & c & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \delta_i & 0 & v_i \\ 0 & \delta_j & v_j \\ h_i & h_j & \alpha \end{pmatrix} \begin{pmatrix} c & \bar{s} & 0 \\ -s & \bar{c} & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} \delta_i & 0 & \bar{c}v_i - \bar{s}v_j \\ 0 & \delta_j & sv_i + cv_j \\ 0 & \tau & \alpha \end{pmatrix}.$$

■

Similarly, if $\delta_i = \delta_j$ and $v_i v_j \neq 0$, M is similar to a matrix with $\delta_i = \delta_j$ and $v_i = 0$. If $h_i = 0$, e_i is the right eigenvector of M corresponding to the eigenvalue δ_i. If $v_i = 0$, e_i is the left eigenvector of M corresponding to the eigenvalue δ_i.

We first consider the case where D has distinct diagonal elements $\delta_1 \neq \delta_2 \ldots \neq \delta_n$ and v and h have no zero elements. By Lemmas 3.1 and 3.2, no diagonal element $\delta_1, \ldots, \delta_{n-1}$ can be an eigenvalue of M. The eigenvalues of M are then the roots of the complex rational equation

$$g(\lambda) = (\lambda - \alpha) + \sum_{i=1}^{n-1} \frac{(e_i{}^T h)(e_i{}^T v)}{\delta_i - \lambda} = 0. \tag{4}$$

The right eigenvector q of M associated with eigenvalue λ is given by

$$q = \begin{pmatrix} \hat{q} \\ \xi \end{pmatrix} = \begin{pmatrix} -(D - \lambda I)^{-1} v \\ 1 \end{pmatrix} \xi, \tag{5}$$

and the left eigenvector is

$$s = \begin{pmatrix} \hat{s} \\ \zeta \end{pmatrix} = \begin{pmatrix} -(D - \lambda)^{-H} \bar{h} \\ 1 \end{pmatrix} \zeta, \tag{6}$$

where ξ, ζ are chosen to make $s_j^H q_j = 1$. (This normalization ensures that the left eigenvector matrix is the inverse of the right eigenvector matrix.)

When there are zero elements in v or h, the matrix $(D - \lambda I)$ is singular, and equations (5) and (6) cannot be used to compute the eigenvectors of M. To derive rules for computing eigenvectors in this case, we first suppose that M has been transformed by a series of elementary transformations accumulated into the matrix G so that equal diagonal elements correspond to zero elements of v or h. We then permute the $n \times n$ matrix $M \leftarrow PGMG^* P^T$ so that all zero elements in its last row or column are grouped together as follows:

$$M = \begin{pmatrix} \hat{D}_0 & 0 & 0 & 0 \\ 0 & \hat{D}_1 & 0 & V \\ 0 & 0 & \hat{D}_2 & 0 \\ 0 & 0 & H & \hat{M} \end{pmatrix}, \tag{7}$$

with

$$H = \begin{pmatrix} 0 \\ \tilde{h}^T \end{pmatrix} \in \mathcal{C}^{r \times p}, \qquad V = \begin{pmatrix} 0 & \tilde{v} \end{pmatrix} \in \mathcal{C}^{t \times r},$$

where $\tilde{v}$ and $\tilde{h}$ have no zero elements. We further assume that the similarity transformations have been applied so that $\hat{D}_1$ and $\hat{D}_2$ have no common diagonal elements. Thus, no eigenvalue of $\hat{D}_1$ is an eigenvalue of $\hat{D}_2$ or of $\hat{M}$.

The zero structure of M allows us to deflate out some eigenpairs. Specifically, The diagonal elements of the diagonal submatrix $\hat{D}_0 \in \mathcal{C}^{j \times j}$ are eigenvalues of the matrix M with corresponding left and right eigenvectors $(I_j,\ 0,\ 0,\ 0)^T$. The diagonal elements of $\hat{D}_1 \in \mathcal{C}^{t \times t}$ are eigenvalues of M with right eigenvectors $(0 \quad I_t \quad 0 \quad 0)^T$. The eigenvalues of $\hat{D}_2 \in \mathcal{C}^{p \times p}$ are eigenvalues of M with corresponding left eigenvectors $(0,\ 0,\ I_p,\ 0)^T$. The remaining eigenvalues of M are the eigenvalues of $\hat{M}$. By Lemmas 3.1 and 3.2, no eigenvalue of $\hat{D}_1$ or $\hat{D}_2$ can also be an eigenvalue of $\hat{M}$.

To compute the remaining eigenvectors, we first determine the eigenvectors of the submatrix

$$M_H = \begin{pmatrix} \hat{D}_2 & 0 \\ H & \hat{M} \end{pmatrix}.$$

The right eigenvector $(\hat{q}_1^T, \hat{q}_2^T)^T$ corresponding to the eigenvalue λ satisfies

$$\begin{aligned} (\hat{D}_2 - \lambda)\hat{q}_1 &= 0 \\ H\hat{q}_1 + (\hat{M} - \lambda)\hat{q}_2 &= 0. \end{aligned}$$

If λ is not a diagonal element of $\hat{D}_2$, then $\hat{q}_1 = 0$, $(\hat{M} - \lambda)\hat{q}_2 = 0$, and $\hat{q}_2$ is the right eigenvector of $\hat{M}$ determined from equation (5). The corresponding left eigenvector $(\hat{s}_1^T, \hat{s}_2^T)^T$ satisfies

$$\begin{aligned} \hat{s}_2^H(\hat{M} - \lambda) &= 0 \\ \hat{s}_1^H(\hat{D}_1 - \lambda) + \hat{s}_2^H H &= 0. \end{aligned}$$

The vector $\hat{s}_2$ is found using equation (6), and the vector $\hat{s}_1$ comes from a diagonal scaling of $s_2^H H$. If λ is the kth diagonal element of $\hat{D}_2$, then $\hat{q}_1 = e_k$, $(\hat{M} - \lambda)\hat{q}_2 = -He_k$, and $\hat{q}_2$ can be found by solving the latter linear system. The corresponding left eigenvector is the appropriate canonical vector. Note that for the arrowhead matrix $\hat{M} \in C^{r \times r}$, any left or right eigenvector of M_H can be found $O(r)$ operations.

We then determine the eigenvectors of the matrix

$$M_V = \begin{pmatrix} \hat{D}_1 & 0 & V \\ 0 & \hat{D}_2 & 0 \\ 0 & H & \hat{M} \end{pmatrix}$$

from those of M_H. The left eigenvector $(s_1^T, s_2^T)^T$ of M_V corresponding to the eigenvalue λ satisfies

$$\begin{aligned} s_1^H(\hat{D}_1 - \lambda) &= 0 \\ s_1^H(0, V) + s_2^H(M_H - \lambda) &= 0 \end{aligned}$$

If λ is not a diagonal element of $\hat{D}_1$, $s_1^H = 0$, and s_2^H is just a left eigenvector of M_H. The corresponding right eigenvector $(q_1^T, q_2^T)^T$ of M satisfies

$$\begin{aligned} (\hat{D}_1 - \lambda)q_1 + (0, V)q_2 &= 0 \\ (M_H - \lambda)q_2 &= 0, \end{aligned}$$

and q_2 is the associated right eigenvector of M_H. The vector q_1 is produced by a diagonal scaling of $(0, V)q_2$. If λ is the kth diagonal element of $\hat{D}_1$, $s_1^H = e_k$, and s_2^H is found by solving the arrowhead system $s_2^H(M_H - \lambda) = -e_k^H(0, V)$ in $O(r + t)$ operations. The corresponding right eigenvector is the appropriate canonical vector.

The eigenvectors of M are those of M_V and of D_0 with zeros appended or prepended as necessary.

4 A tearing for Hessenberg matrices

In this section, we show how the divide and conquer scheme of Section 3 can be applied to an upper Hessenberg matrix H of order $n = 2m+1$, under the assumption that the Hessenberg submatrices formed by matrix tearing are diagonalizable. If $H_1 = X_1 D_1 X_1^{-1}$ and $H_2 = X_2 D_2 X_2^{-1}$, then

$$
\begin{aligned}
H &= \left(\begin{array}{c|c|c} H_1 & & Z \\ \hline & \alpha_m & \\ \hline & & H_2 \end{array}\right) + \left(\begin{array}{c|c|c} & v & \\ \hline \gamma_m & & h^T \\ \hline & \gamma_{m+1} & \end{array}\right) \\
&= X \left[\left(\begin{array}{c|c|c} D_1 & & \\ \hline & \alpha & \\ \hline & & D_2 \end{array}\right) + \left(\begin{array}{c|c|c} & X_1^{-1}v & X_1^{-1}ZX_2 \\ \hline \gamma_m e_m^T X_1 & & h^T X_1 \\ \hline & \gamma_{m+1}X_2^{-1}e_1 & \end{array}\right) \right] X^{-1}
\end{aligned}
$$

with $\alpha = \alpha_m$ and X the direct sum $X_1 \oplus 1 \oplus X_2$. The interior matrix can be permuted to form

$$
M = \left(\begin{array}{c|c|c} D_1 & X_1^{-1}ZX_2 & v_1 \\ \hline & D_2 & v_2 \\ \hline h_1^T & h_2^T & \alpha \end{array}\right), \tag{8}
$$

where $v_1 = X_1^{-1}v$, $v_2 = \gamma_{m+1}X_2^{-1}e_1$, $h_1 = \gamma_m X_1^T e_m$, and $h_2 = X_2^T h$. The eigenvalues of M not equal to diagonal elements of D_1 and D_2 are the roots of the secular equation

$$
\begin{aligned}
r(\lambda) \;=\; & (\alpha - \lambda) + h_1^T(D_1 - \lambda I)^{-1}v_1 + h_2^T(D_2 - \lambda I)^{-2}v_2 - \\
& h_1^T(D_1 - \lambda I)^{-1}(X_1^{-1}ZX_2)(D_2 - \lambda I)^{-1}v_2.
\end{aligned} \tag{9}
$$

Note that $r(\lambda) = 0$ has the same form as the tridiagonal secular equation (4) plus an additional cross term involving Z.

The right eigenvector of M for eigenvalue λ is

$$
q = \begin{pmatrix} q_1 \\ q_2 \\ \zeta \end{pmatrix} = \zeta \begin{pmatrix} -(D_1 - \lambda I)^{-1}[-(X_1^{-1}ZX_2)(D_2 - \lambda I)^{-1}v_2 + v_1] \\ -(D_2 - \lambda I)^{-1}v_2 \\ 1 \end{pmatrix}.
$$

The deflation rules for the right eigenvectors of M derive from this expression (using $Mq = \lambda q$) and are not as simple as those given for the tridiagonal case in Lemmas 3.1 and 3.2. For example, a diagonal element of D_1 is retained as an eigenvalue λ of M whenever $D_2 - \lambda I$ is nonsingular, and both of the following relations are satisfied:

$$
\begin{aligned}
\zeta e_j^T v_1 &= \zeta e_j^T (X_1^{-1}ZX_2)(D_2 - \lambda I)^{-1}v_2 \\
h_1^T q_1 &= \zeta \left[h_2^T(D_2 - \lambda I)^{-1}v_2 + (\alpha - \lambda) \right]
\end{aligned}
$$

for $e_j^T(D_1 - \lambda I) = 0$ and any choice of q_2. The requirements are similarly complicated if λ is a diagonal element of D_2 but not of D_1. If λ appears in both D_1 (jth diagonal element) and D_2 (kth diagonal element), the requirements simplify to

$$
\begin{aligned}
-\zeta e_j^T v_1 &= e_j^T(X_1^{-1}ZX_2)q_2 \\
\zeta e_k^T v_2 &= 0
\end{aligned}
$$

for any q_1. The formulas for the left eigenvector and deflation rules resemble those for the right eigenvector. The divide and conquer mechanism thus extends to the Hessenberg case, although less simply and less efficiently than to the tridiagonal case.

5 Obstacles to the Nonsymmetric Method

To this point, we have shown that as long as the submatrices formed from tearing are diagonalizable, a divide and conquer method for nonsymmetric tridiagonal and Hessenberg eigenproblems can be derived along the lines of the symmetric method reviewed in Section 2. In the remainder of this paper, we discuss difficulties with efficiency and stability that stand in the way of a practical implementation of a nonsymmetric divide and conquer method. We use our method and that of [1] to illustrate the problems, but many of our deliberations would apply to any divide and conquer algorithm of this type.

5.1 Root-finding

An accurate and efficient root-finder is essential to the success of the symmetric divide and conquer method of [8, 12]. The nonsymmetric tridiagonal secular equation (4) $g(\lambda) = 0$, however, shares few of the properties that make the symmetric secular equation (2) $w(\lambda) = 0$ so easy to solve. The nonsymmetric function $g(\lambda)$ can still be evaluated in $O(n)$ flops, but because it derives from the spectral decompositions of T_1 and T_2, it can have complex coefficients. The complex roots of $g(\lambda) = 0$ occur in conjugate pairs but otherwise can lie anywhere within the union of Gerschgorin disks of T or of M. Only the poles $\delta_1, \ldots, \delta_{n-1}$ of $g(\lambda)$ are easily identified. Unlike the symmetric equation, where the roots interlace the poles, there is no obvious connection between the locations of the poles and roots of $g(\lambda) = 0$. The only advantage of the nonsymmetric equation is that only its roots with nonnegative (or nonpositive) imaginary roots need be explicitly computed to determine the full spectrum.

To illustrate the general structure of $g(\lambda)$ we present contour plots of $\log(|g(\lambda)|)$ over the area of the complex plane containing all roots and poles for two 5×5 matrices. Figure 1 shows the function for matrix T_5 having diagonal elements 1, 2, 3, 4, 5 and off-diagonal elements $\beta_k = 1$, $\gamma_k = -1$, $k = 1, \ldots, 4$. Boxes mark the four complex poles of the function, and $\times$'s mark its one real and four complex roots. In this case, the structure of $|g(\lambda)|$ is quite regular. Figure 2 shows $\log(|g(\lambda)|)$ for a 5×5 random tridiagonal matrix. In this instance, two poles and three roots lie on the real line. The remaining complex conjugate poles and roots are nearly coincident, although deflation ensures that they are not equal. (In both cases, the function rises monotonically toward the poles: apparent structure near the poles is an artifact of the printer's resolution.)

In this section, we examine the possibility of unconditional global convergence to all roots of $g(\lambda) = 0$. As the roots and poles of this equation do not interlace, the first order of business is to localize the roots. To this end, we present Weyl's algorithm [20] which was originally devised for polynomials. Given an initial search interval for the roots (here, the union of all Gerschgorin disks of T or of the deflated matrix M), we can cover it with closed squares and determine the number of roots lying within each. Any squares found to contain no roots are discarded from the search area. The remaining squares are subdivided and the process is repeated recursively with the smaller squares. (Lehmer's method uses a similar process that covers the search area with disks [20, 21]. Derwidué's method covers the upper half plane with strips [21].)

To implement Weyl's algorithm, it remains to devise an *exclusion test* to determine which squares are empty [20]. Let Γ be the boundary of a square to be tested. Because we know the locations of the poles of $g(\lambda)$, we could determine the number of roots inside of Γ by numerically evaluating the winding number

$$n = \frac{1}{2\pi i} \int_{\Gamma} \frac{g'(\lambda)}{g(\lambda)} d\lambda, \tag{10}$$

which equals the number of zeros of $g(\lambda)$ inside Γ minus the number of poles of $g(\lambda)$ inside Γ [7]. The cost of computing this integral can be high, however, and it may be impossible to attain an accurate result when zeros lie close to Γ [20]. An alternative is to see whether $\max_{\Gamma} |g(\lambda)|$ ever exceeds $\max_{\Gamma} |g'(\lambda)| \sqrt{\omega}$ on a square Γ of side ω not enclosing a pole. If it does, the function cannot have a zero inside of Γ [7]. (A similar test is described in [20] for polynomials.) Determining these maxima amounts to yet another root-finding or optimization problem and so would be expensive. Such a test could also fail for roots near poles or near the edge of a square.

Even if a practical exclusion test could be devised, Weyl's method offers only linear convergence. Furthermore, Weyl's algorithm requires that significant work be performed on intervals containing no eigenvalues. (In the symmetric case, the interlacing property means that only intervals of the real line containing eigenvalues are examined [12, 22].) Weyl's algorithm does lend itself to a parallel implementation as each square can be examined independently by one processor, although experiments with one-dimensional multisection routines suggest that an efficient mechanism for dynamically assigning squares to processors would be required for good performance [22, 26]. According to Henrici [20], this sort of multisection procedure is the only way to compute all roots of a polynomial to a desired precision with unconditional global convergence. We are not aware of any other globally convergent method for computing all zeros of rational functions.

While it appears that there is no reasonably efficient globally convergent method for solving the secular equation, alternatives such as Newton's method, Bernoulli's method, Graeffe's root squaring method, or Rutishauser's quotient-difference method [20, 21] might work in practice. We could also pose the root-finding problem as the problem of finding all local minimizers of $|g(\lambda)|$ within the minimal set of Gerschgorin disks of T or M. As a consequence of the Maximum Modulus Theorem [7], the modulus $|g(\lambda)|$ has no local minimizers other than its roots. In this case, we can use the method of steepest descent for guaranteed convergence to a single root [17, 20]. Any of these methods would require a mechanism for choosing starting guesses for each root and a mechanism for determining if all roots have been computed.

Finally, we note that the $O(n)$ cost for evaluating $g(\lambda)$ results from having a *tridiagonal* matrix torn into *diagonalizable* submatrices. In contrast, when evaluating the Hessenberg secular equation (9), we must first determine $X_1^{-1} Z X_1$ in $O(m^3)$ operations then evaluate $r(\lambda) = 0$ in $O(m^2)$ operations where $m = \frac{n-1}{2}$. If the submatrices resulting from tearing a tridiagonal or Hessenberg matrix, are not diagonalizable, the cost of evaluating the resulting secular equation is also higher than for $g(\lambda)$ or $r(\lambda)$, respectively [1].

5.2 Eigenvector Computation

The derivation in Section 3 shows that, for an $n \times n$ matrix M, all left and right eigenvectors are produced by our nonsymmetric divide and conquer algorithm in $O(n^2)$ operations. Furthermore, at each stage, every eigenvector of M_H, M_V, or M is computed independently of every other. Although independent computation of eigenvectors is good for parallel efficiency, it may not be the best numerical strategy for the nonsymmetric eigenproblem. To compute an accurate singular value decomposition of a matrix by the divide and conquer strategy of [24], it is necessary to compute each right singular vector from its corresponding left singular vector rather than by independent formulas. In part, this ensures correct pairing of left and right singular vectors corresponding to close singular values. Similarities between the divide and conquer algorithms for the SVD and the nonsymmetric eigenproblem suggest than an accurate eigensolver for the latter might also require the right eigenvectors to be computed from the left eigenvectors.

In the SVD algorithm of [24], the right singular vector is given by a diagonal scaling of the left singular vector. The relationship between the left and right singular vectors that makes this efficient transformation possible does not carry through to the nonsymmetric eigenproblem. If the matrix $\hat{M}$ in equation (7) is diagonalizable, the matrix of right eigenvectors Q can be computed from the left eigenvectors only by inverting the latter in $O(r^3)$ operations. If the eigendecomposition $M_H = Q_H \Lambda_H Q_H^{-1}$ is computed stably, the left eigenvectors of the diagonalizable matrix M_V derive from

$$M_V = \begin{pmatrix} \hat{D}_1 & V \\ 0 & M_H \end{pmatrix}$$

and the relation

$$\Lambda = \begin{pmatrix} \hat{D}_1 & V \\ 0 & \Lambda_H \end{pmatrix} = \begin{pmatrix} I & A \\ 0 & Q_H^{-1} \end{pmatrix} \begin{pmatrix} \hat{D}_1 & 0 \\ 0 & M_H \end{pmatrix} \begin{pmatrix} I & C \\ 0 & Q_H \end{pmatrix}. \tag{11}$$

In this case, $A = -CQ_H^{-1}$, and each element of C is computed by

$$e_j^T C e_k = -(e_j^T V)(Q_H^{-1} e_k)/(e_j^T \hat{D}_1 e_j - e_k^T \Lambda_H e_k),$$

$1 \leq j \leq p$ and $1 \leq k \leq r+t$. Computing the submatrix C takes $O(p(r+t))$ operations, but computing A takes $O(p(r+t)^2)$ operations. The matrix product needed to compute A and the inversion of Q^{-1} make the eigenvector computation expensive both serially and in parallel. In the algorithm of [1], the left and right eigenspaces are computed independently.

5.3 Deflation

In practice, the cost of the symmetric divide and conquer method of [8, 12] is strongly problem dependent and is determined by the amount of deflation occurring for a given problem [12, 23]. When deflation is prevalent, divide and conquer is the fastest way to compute accurately all eigenvalues and eigenvectors of a symmetric tridiagonal matrix, but if no deflation occurs, the divide and conquer and QR methods are of comparable cost [23]. We have not done experiments to determine the likelihood of deflation in the nonsymmetric case but do note that the zero structure of the computed eigenvector matrices could influence the savings. For symmetric matrices, the eigenvector matrix decouples into the direct sum of an identity matrix and a computed submatrix, but for nonsymmetric matrices, the eigenvector matrices have nonsymmetric zero structure and, hence, possibly fewer zeros than if $v = h$. Fewer zeros would mean less savings when the eigenvectors of M are computed and and when they are backtransformed to those of the original matrix T.

As in the symmetric case, deflation rules can be formulated for use in a finite precision implementation. A diagonal element δ_j of the matrix M is retained as an eigenvalue of M when δ_j is *close* to some other diagonal element δ_i or when an element v_i or h_i is small. In the first instance, similarity transformations are applied as in Lemma 3.2 to zero out a border element h_j or v_j corresponding to one duplicate diagonal element. Error analysis of this process proceeds much as for the symmetric case in [6, 12] and shows that finite precision deflation leads us to work with a matrix $M + E$ close to the one we would use in exact arithmetic.

If we accept the eigendecomposition of $M + E$ as the eigendecomposition of M, we can then see an error in a computed eigenvalue λ_i of M as large as [19, 31]

$$|\lambda_i(M) - \lambda_i(M+E)| \approx \frac{|E|}{\sigma(\lambda_i(M))},$$

where $1/\sigma(\lambda_i)$ is the condition number of the eigenvalue λ_i. Thus, if all eigenvalues of M are well conditioned, deflation of this sort should pose no threat to the accuracy of the method. Similarly, a well-conditioned eigenvector would not be sensitive to the small errors introduced by the similarity transformations.

Ignoring small elements of h and v also appears to have little influence on the accuracy of the solution as measured by the residual error. We accept (δ_i, e_i) as a right eigenpair of M whenever $|h_i| \leq \text{tol} = \theta \epsilon_M \| M \|$ with $\theta = O(1)$ and ϵ_M = machine epsilon. This ensures that the residual error of the eigenpair is bounded above by a small value

$$\| Me_i - \delta_i e_i \| = |(h^T e_i) e_n| = |h_i| \leq \text{tol}.$$

These results indicate that deflation should not influence the accuracy of well-conditioned eigenpairs but could be problematic for ill-conditioned eigenpairs.

The time savings due to deflation are difficult to predict. The nonsymmetric zero structure of the eigenvector matrices leads to reduced savings in both computation and backtransformation of the eigenvectors of M when compared to deflation in the symmetric method. The loss of the efficient root-finder, however, suggests that eigenvalue computation time may occupy a substantially larger fraction of the total time than it does in the symmetric case. If this is true, the savings in eigenvalue computation following deflation would be more important for nonsymmetric than for symmetric matrices.

The algorithm of [1] attains a low operation count by computing the eigenvectors of T through direct system solution rather than by computing and backtransforming the eigenvectors of an intermediate matrix. Adams and Arbenz conclude that as long as the required system solution is stable, deflation of the problem is unnecessary. (The stability of the computation is as yet unestablished.)

5.4 Stability

There remains one additional aspect of a divide and conquer method that can most severely impact its accuracy even if it allows for defective submatrices. Namely, if either of the torn submatrices T_1 or T_2 of a nonsymmetric matrix T is ill-conditioned with respect to the eigenproblem, the accuracy of its computed eigendecomposition can be poor [19, 31]. Because the eigendecomposition of T is built up from those of T_1 and T_2 and is never corrected using T itself, the eigendecomposition of T will also be inaccurate.

The following example demonstrates that even a matrix that is well-conditioned for eigendecomposition can have ill-conditioned submatrices. The Hessenberg matrix

$$H(\delta) = \begin{pmatrix} 0 & 0 & 0 & 10^{-8}+\delta & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix}$$

has distinct eigenvalues all with condition numbers near 10 and an eigenvector matrix with 2-norm condition number of 3.4 when $\delta = 0$. Like $H(0)$, the 4×4 submatrix formed by rank-two tearing

$$H_2 = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix}$$

has distinct eigenvalues with condition numbers near 10 and an eigenvector matrix with 2-norm condition number 2.0. However, the submatrix

$$H_1(\delta) = \begin{pmatrix} 0 & 0 & 0 & 10^{-8}+\delta \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

has four distinct eigenvalues with condition numbers around $10^5 - 10^6$ and an eigenvector condition number of about 10^6 for δ near or equal to zero. (See [31].)

To demonstrate the sensitivity of $H(\delta)$ to the ill-condition of $H_1(\delta) = X_1(\delta)D_1(\delta)X_1(\delta)^{-1}$ in our divide and conquer algorithm, we first compute the structured matrix

$$M(\delta) \quad = \quad \left(\begin{array}{c|c|c} D_1(\delta) & X_1(\delta)^{-1}ZX_2 & v_1(\delta) \\ \hline & D_2 & v_2 \\ \hline h_1(\delta)^T & h_2^T & \alpha \end{array}\right)$$

from equation (8) using Matlab [27] for $\delta = 0$ and $\delta = \epsilon_M = 2.22 \times 10^{-16}$. All eigendecompositions needed to construct and analyze M are computed in double precision using the Matlab function *eig* which computes the eigenvalues by the QL method [29, 27].

The small perturbation in $H(\delta)$ leads to a difference in $M(\delta)$ of

$$\| \, M(\epsilon_M) - M(0) \, \|_\infty = 0.14$$

or a relative difference of

$$\| \, M(\epsilon_M) - M(0) \, \|_\infty / \| \, M(0) \, \|_\infty = 2.28 \times 10^{-7}.$$

This change in M leads to the following relative changes in the computed eigendecomposition of $M(\delta) = Q(\delta)\Lambda(\delta)Q(\delta)^{-1}$:

$$\begin{aligned}
\| \Lambda(\epsilon_M) - \Lambda(0) \|_\infty / \| \Lambda(0) \|_\infty &= 1.89 \times 10^{-11} \\
\| Q(\epsilon_M) - Q(0) \|_\infty / \| Q(0) \|_\infty &= 1.10 \\
\| Q(\epsilon_M)^{-1} - Q(0)^{-1} \|_\infty / \| Q(0)^{-1} \|_\infty &= 0.32.
\end{aligned}$$

In the latter cases, the absolute errors are

$$\begin{aligned}
\| Q(\epsilon_M) - Q(0) \|_\infty &= 1.62 \times 10^3 \\
\| Q(\epsilon_M)^{-1} - Q(0)^{-1} \|_\infty &= 0.025.
\end{aligned}$$

The residual error for the computed right eigenpairs is

$$\| M(0)Q(\epsilon_M) - Q(\epsilon_M)\Lambda(\epsilon_M) \|_\infty / \| \Lambda(\epsilon_M) \|_\infty = 3.85 \times 10^{-4}.$$

By any of these measurements, we see that a very small change in $H(\delta)$ leads to a significantly larger error in its computed eigendecomposition. An ill-conditioned problem in which deflation takes place could be even more inaccurate. As these errors are caused by ill-condition, there is no reason to expect a substantially different result when the QL method used for our experiments is replaced by root-finding.

If an inaccurate subproblem solution is detected, the matrix could be divided at a different point and the ill-conditioned submatrix replaced with a new submatrix. Despite the additional work involved, this approach cannot guarantee new subproblems with better condition than the original ones. Furthermore, efficient divide and conquer methods typically recursively subdivide the original matrix more than once to form a tree of updating problems [12, 22, 24]. Even if the smallest submatrices at the leaves of the tree are well-conditioned with respect to the eigenproblem, larger submatrices farther up the tree may not be. To correct for ill-conditioning at level k of the tree, it would be necessary to redivide the matrix and repeat the updating procedure for all levels of the tree from the leaves to level k. Again, this redivision could not guarantee improved condition of the level k problems. To make matters worse, ill-conditioning could be introduced in the new problems below level k.

6 Conclusion

We have seen that the divide and conquer method that has been so successfully applied to other matrix problems can fall short when extended to the nonsymmetric eigenvalue problem. The speed and accuracy of the other methods rely largely on the availability of a fast and globally convergent root-finder and on the prevalence and ease of deflation. However, there appears to be no equivalent root-finder for the nonsymmetric case, and deflation of of the nonsymmetric problem may not be as advantageous as in the symmetric case. Furthermore, if it is necessary to compute the left eigenvectors from the right eigenvectors to maintain accuracy, the eigenvector computation may become inefficient, especially in parallel.

The greatest danger with the divide and conquer method, however, lies in its potential instability. Even if the original matrix is well-conditioned with respect to the eigenproblem, an ill-conditioned submatrix can be created at any level of updating. Thus, even small errors introduced by tearing, deflation, or updating can lead to large errors in the computed eigendecomposition. Because the original matrix is never used in the updating procedure, there is no opportunity to correct an error introduced by an ill-conditioned submatrix. Two divide and conquer methods that employ the matrix original matrix T in the updating procedure and appear to overcome poor intermediate results are discussed in [11, 25].

7 Acknowledgements

The author thanks Loyce Adams, Peter Arbenz, Jack Dongarra, and Majed Sidani for several helpful discussions and thanks Bob Manchek for preparing Figures 1 and 2.

References

[1] L. Adams and P. Arbenz. Towards a divide and conquer algorithm for the real nonsymmetric eigenvalue problem. Department of Applied Mathematics, University of Washington, Research Report Preprint.

[2] G.S. Ammar, D. Cheng, W. Dayawansa, and C. Martin. Identification of linear systems by Prony's method. In *Robust Control of Linear Systems and Nonlinear Control*, pages 483–488, 1990.

[3] G.S. Ammar, W. Dayawansa, and C. Martin. Exponential interpolation: theory and numerical algorithms. To appear in J. Appl. Math. Comput.

[4] G.S. Ammar, L. Reichel, and D.C. Sorensen. An implementation of a divide and conquer method for the unitary eigenproblem. To appear in ACM TOMS.

[5] J. Barlow. Error analysis of update methods for the symmetric eigenvalue problem. To appear in SIAM J. Matrix Anal. Appl.

[6] J.R. Bunch, C.P. Nielsen, and D.C. Sorensen. Rank-one modification of the symmetric eigenproblem. *Numer. Math.*, 31:31–48, 1978.

[7] G.F. Carrier, M. Krook, and C.E. Pearson. *Functions of a Complex Variable.* McGraw-Hill Book Company, 1966.

[8] J.J.M. Cuppen. A divide and conquer method for the symmetric tridiagonal eigenproblem. *Numer. Math.*, 36:177–95, 1981.

[9] J. Dongarra, G.A. Geist, and C.H. Romine. Computing the eigenvalues and eigenvectors of a general matrix by reduction to tridiagonal form. Technical Report ORNL/TM-11669, Oak Ridge National Laboratory, 1990.

[10] J. Dongarra, S. Hammarling, and D. Sorensen. Block reduction of matrices to condensed form for eigenvalue computations. *Journal of Computational and Applied Mathematics*, 27:215–227, 1989.

[11] J. Dongarra and M. Sidani. A divide and conquer method for the nonsymmtric eigenvalue problem. Department of Computer Science, University of Tennessee, Research report in preparation.

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

[13] K. Gates. University of Washington, PhD thesis in preparation, 1991.

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

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

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

[17] P.E. Gill, W. Murray, and M.H. Wright. *Practical Optimization.* Academic Press, 1981.

[18] G.H. Golub. Some modified matrix eigenvalue problems. *SIAM Review*, 15:318–34, 1973.

[19] G.H. Golub and C.F. Van Loan. *Matrix Computations.* The Johns Hopkins Press, Baltimore, MD, 2nd edition, 1989.

[20] P. Henrici. *Applied and Computational Complex Analysis.* John Wiley and Sons, 1974.

[21] A.S. Householder. *The Numerical Treatment of a Single Nonlinear Equation.* McGraw-Hill, 1970.

[22] I.C.F. Ipsen and E.R. Jessup. Solving the symmetric tridiagonal eigenvalue problem on the hypercube. *SIAM J. Sci. Stat. Comput., Vol. 11, No. 2*, pages 203–229, 1990.

[23] E.R. Jessup and I.C.F. Ipsen. Improving the accuracy of inverse iteration. To appear in SIAM J. Sci. Stat. Comput.

[24] E.R. Jessup and D.C. Sorensen. A parallel algorithm for computing the singular value decomposition of a matrix. Technical Report ANL/MCS-TM-102, Argonne National Laboratory, 1987.

[25] T.Y. Li, Z. Zeng, and L. Cong. Solving eigenvalue problems of real nonsymmetric matrices with real homotopies. Dept. Mathematics and Dept. Computer Science, Michigan State University, Research Report Preprint, 1990.

[26] S. Lo, B. Phillipe, and A. Sameh. A multiprocessor algorithm for the symmetric tridiagonal eigenvalue problem. *SIAM J. Sci. Stat. Comput.*, 8:s155–s165, 1987.

[27] C. Moler, J. Little, S. Bangert, and S. Kleiman. *Pro-Matlab User's Guide.* The MathWorks, Inc., 1987.

[28] D.P. O'Leary and G.W. Stewart. Computing the eigenvalues and eigenvectors of symmetric arrowhead matrices. *J. Comp. Phys.*, 90:497–505, 1990.

[29] B.T. Smith, J.M. Boyle, J.J. Dongarra, B.S. Garbow, Y. Ikebe, V.C. Klema, and C.B. Moler. *Matrix Eigensystem Routines–EISPACK Guide, Lecture Notes in Computer Science, Vol. 6, 2nd edition.* Springer-Verlag, 1976.

[30] D.C. Sorensen and P. Tang. On the orthogonality of eigenvectors computed by divide and conquer techniques. To appear in SIAM J. Numerical Analysis.

[31] J.H. Wilkinson. *The Algebraic Eigenvalue Problem.* Clarendon Press, Oxford, 1965.

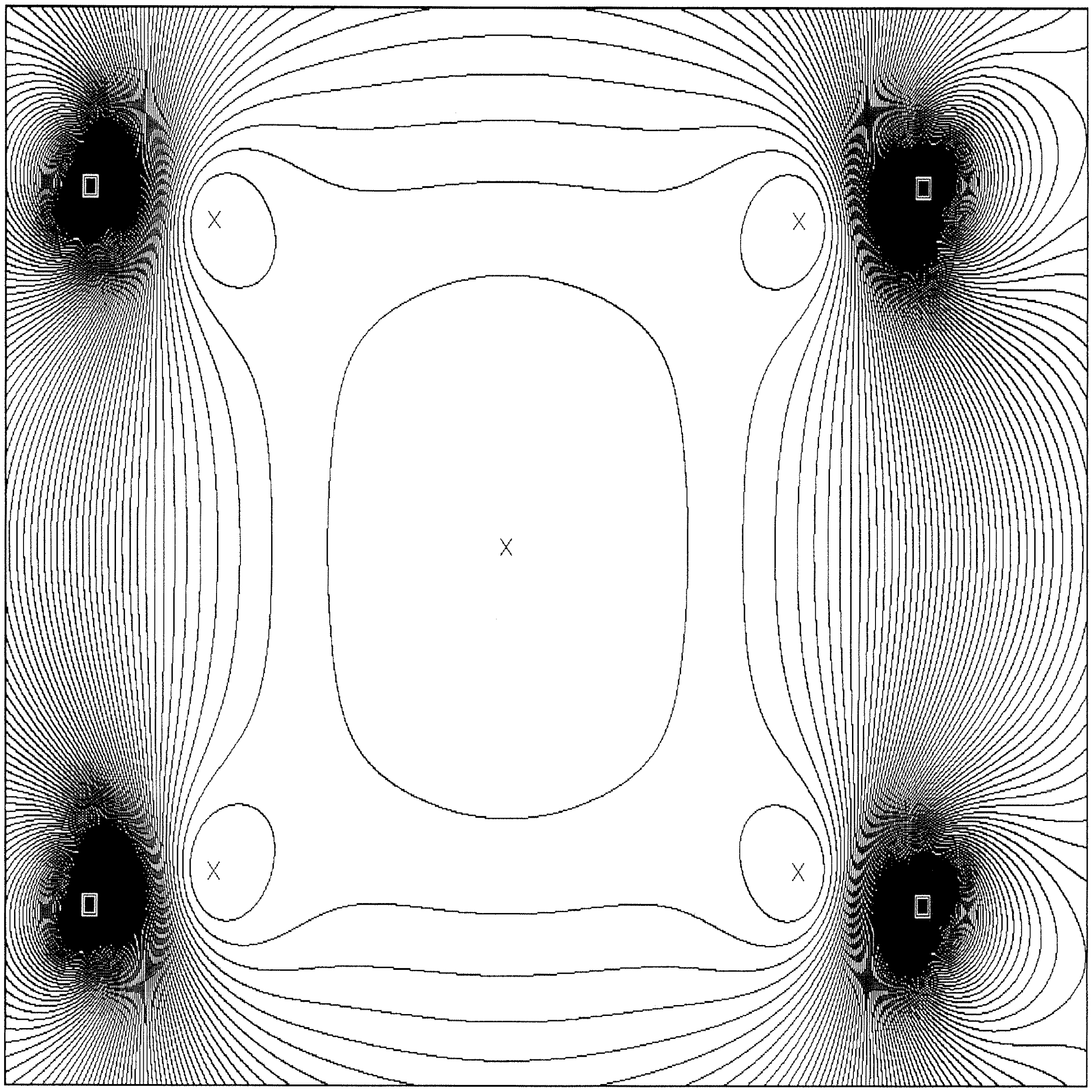

Figure 1: Contour plot of $\log(|g(\lambda)|)$ for the 5×5 matrix T_5. Poles are marked with boxes, and roots are marked with $\times$'s

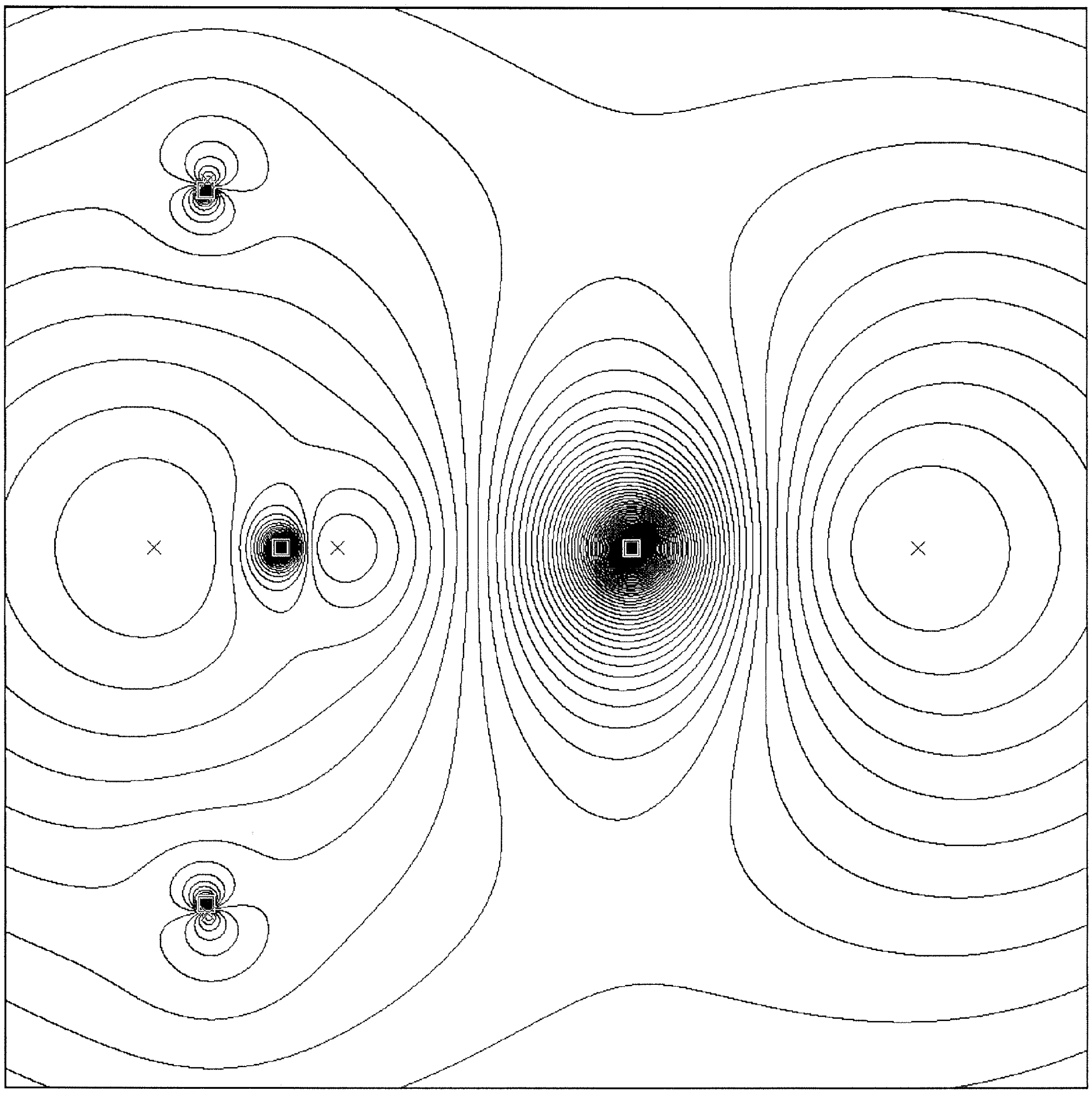

Figure 2: Contour plot of $\log(|g(\lambda)|)$ for a random 5×5 matrix. Poles are marked with boxes, and roots are marked with $\times$'s