Towards a divide and conquer algorithm for the real nonsymmetric eigenvalue problem

Eidgenössische
Technische Hochschule
Zürich

Departement Informatik
Institut für
Wissenschaftliches Rechnen

Loyce Adams
Peter Arbenz

Towards a Divide and Conquer Algorithm for the Real Nonsymmetric Eigenvalue Problem

September 1991

165

This report appeared also as Technical Report No. 91-8, University of Washington, Department of Applied Mathematics, Seattle, WA 98195, USA, August 1991.

Authors´ address:

Prof. Loyce M. Adams
Departement of Applied Mathematics, FS-20
University of Washington
Seattle, WA 98195, USA
e-mail: adams@amath.washington.edu

Dr. Peter Arbenz
Institut für Wissenschaftliches Rechnen
ETH Zentrum
CH-8092 Zurich, Switzerland.
e-mail: arbenz@inf.ethz.ch

TOWARDS A DIVIDE AND CONQUER ALGORITHM FOR THE REAL NONSYMMETRIC EIGENVALUE PROBLEM

LOYCE ADAMS * AND PETER ARBENZ †

Abstract. Theory is developed that could be used towards developing a divide and conquer algorithm for the nonsymmetric eigenvalue problem. We discuss the shortcomings of this theory and discuss its application to the Hessenberg and nonsymmetric tridiagonal problems. We conclude that the method may not be as promising as the divide and conquer methods for symmetric problems.

Key words. tridiagonal matrix, Hessenberg matrix, modified eigenvalue problem, divide and conquer algorithm.

AMS(MOS) subject classifications. 15A18, 65F15, 65W05.

1. Introduction. In 1981, Cuppen [7] introduced a divide and conquer algorithm for the computation of the spectral decomposition of real symmetric tridiagonal matrices. The algorithm which was primarily designed for parallel computers turned out to be faster than and comparably accurate to the well-known QR-algorithm even on sequential computers [8]. We will review this algorithm in §2. Similar divide and conquer algorithms have proven efficient for the bidiagonal singular value problem [16] and the unitary eigenvalue problem [1].

In this paper we investigate whether it is possible to derive an algorithm similar to Cuppen's to solve the eigenvalue problem for real nonsymmetric matrices. More generally we investigate how far the theory for symmetric low rank modified eigenvalue problems as developed in [3] carries over to the nonsymmetric case. The theory is then applied to two special cases of eminent interest: The eigenvalue problem for real (upper) Hessenberg matrices, and for nonsymmetric tridiagonal matrices is solved by a divide and conquer algorithm based on rank-1 modified eigenvalue problems. Jessup [15] has recently investigated the nonsymmetric Hessenberg and tridiagonal problems by using a rank-2 splitting assuming the matrices are diagonalizable.

The interest for the Hessenberg form stems from the fact that general matrices are transformed into this form before the QR algorithm is applied. We note here that the QR algorithm computes a Schur decomposition. In contrast, divide and conquer algorithms work with eigenvectors. They therefore provide a spectral decomposition or, in the case of defective matrices, a partial spectral decomposition.

The nonsymmetric tridiagonal eigenvalue problem arises e.g. in connection with the nonsymmetric Lanczos algorithm [13, p.502]. It is also possible to transform a general matrix into a similar tridiagonal matrix by means of (nonorthogonal) Householder elementary reflectors. This process however is not stable [20, p.403]. Nevertheless, efforts have been made to stabilize it (see [11] and the references therein).

The paper is organized as follows. After the review of Cuppen's algorithm, we derive in section 3 a general theory for restricted rank modified eigenvalue problems of the form $\tilde{A} = A + UV^T$, where U and V are real $n \times r$ matrices of maximal rank. The

* Department of Applied Mathematics, University of Washington, Seattle WA 98195. The work of this author was partially supported by NSF PYI ASC-8858101 and DOE grant DE-FG06-88ER25061 and was completed while visiting the Institut für Wissenschaftliches Rechnen, ETH Zürich, Switzerland.

† ETH Zentrum, Institut für Wissenschaftliches Rechnen, 8092 Zürich, Switzerland.

theory shows how eigenvalues together with their geometric and algebraic multiplicity, eigenvectors and principal vectors of $\tilde{A}$ can be found, if eigenvalues and eigenvectors of A are known. We also discuss what would have to be done to get Schur vectors. In section 4 we discuss deflation for the general rank-r and the practically important rank-1 case. In section 5 we rewrite the above theory for a particular representation of A. In section 6 we apply our theory to the cases where A are tridiagonal and Hessenberg matrices, respectively. We are able to formulate algorithms to solve these cases and to estimate their serial as well as parallel complexities. We discuss the probable behavior of the algorithms in the final section 7.

2. Cuppen's Divide and Conquer Algorithm. In this section we consider the problem of determining the spectral decomposition

$$\tilde{T} = \tilde{X}\tilde{\Lambda}\tilde{X}^T, \qquad \tilde{T}, \tilde{X}, \tilde{\Lambda} \in \mathbb{R}^{n\times n}, \tag{2.1}$$

of the symmetric tridiagonal matrix

$$\tilde{T} = \begin{pmatrix} \alpha_1 & \beta_1 & & \\ \beta_1 & \alpha_2 & \ddots & \\ & \ddots & \ddots & \beta_{n-1} \\ & & \beta_{n-1} & \alpha_n \end{pmatrix} \in \mathbb{R}^{n\times n}.$$

Cuppen [7] introduced the decomposition

$$\begin{aligned} \tilde{T} &= T + D = \begin{pmatrix} T_1 & 0 \\ 0 & T_2 \end{pmatrix} + \beta_k \begin{pmatrix} 0 & & & & & \\ & \ddots & & & & \\ & & 0 & & & \\ & & & \theta & 1 & \\ & & & 1 & \theta & \\ & & & & & 0 \end{pmatrix} \\ &= T_1 \oplus T_2 + \theta\beta_k u u^T, \end{aligned} \tag{2.2}$$

where $u = e_k + \theta e_{k+1}$ with $\theta = \pm 1$. e_j denotes the j-th unit vector. The introduction of the factor θ was indeed an idea of Dongarra and Sorensen [8] to avoid cancellation when forming the new diagonal elements of T_1 and T_2.

As the element of T at position $(k, k+1)$ vanishes, the computation of its spectral decomposition amounts to the solution of two *independent* symmetric tridiagonal eigenvalue problems for T_1 and T_2 of order k and $n-k$, respectively. With the spectral decompositions $T_i = X_i\Lambda_i X_i^T$, $i = 1, 2$, we get

$$\tilde{T} = X\left[\Lambda + \theta\beta_k v v^T\right] X^T \tag{2.3}$$

where

$$X = X_1 \oplus X_2, \quad \Lambda = \Lambda_1 \oplus \Lambda_2, \quad v = X^T u = \begin{pmatrix} X_1^T e_k \\ \theta X_2^T e_1 \end{pmatrix}.$$

Thus, if we know Λ_1 and Λ_2, we obtain the eigenvalues and vectors of $\tilde{T}$ by computing the spectral decomposition of the matrix in square brackets i.e. the spectral decomposition of a diagonal modified by a matrix of rank one.

As the matrices $(\tilde{T}-\lambda I)\oplus 1$ and $(T-\lambda I)\oplus(1+\theta\beta_k u^T(T-\lambda I)^{-1}u)$ for $\lambda\notin\sigma(T)$ are congruent [2],[3],[4], the eigenvalues and vectors of $\tilde{T}$ can be obtained from the ones of T by an investigation of the rational function [2],[5],[7],[8],[12],[14],[20]

$$\begin{aligned} f(\lambda) &= 1+\theta\beta_k u^T(T-\lambda I)^{-1}u = 1+\theta\beta_k v^T(\Lambda-\lambda I)^{-1}v \\ &= 1+\theta\beta_k\sum_{i=1}^{n}\frac{\nu_i^2}{\lambda_i-\lambda} = \frac{\det(\tilde{T}-\lambda I)}{\det(T-\lambda I)}, \qquad v=\{\nu_i\}_{i=1}^n . \end{aligned} \tag{2.4}$$

f is called the (modified) Weinstein determinant [14],[19]. Alternatively, $f(\lambda)=0$ is called the secular equation [12]. By (2.4) one easily obtains the interlacing properties

$$\lambda_j\le\tilde{\lambda}_j\le\lambda_{j+1},\quad 1\le j<n,\qquad \lambda_n\le\tilde{\lambda}_n \tag{2.5}$$

for positive $\theta\beta_k$. (Similar inequalities hold if $\theta\beta_k<0$.) Note that the λ_j do not appear ordered in the diagonal of Λ!

From (2.3) it is seen that the eigenvalue λ_j persists (with unchanged eigenvectors) if $\nu_j=0$. In this case we can *deflate* i.e. remove the corresponding j-th row and column from $\Lambda+\theta\beta_k vv^T$. Moreover, if λ is an eigenvalue of T of multiplicity $m>1$, by an orthogonal similarity transformation of (2.3) the corresponding eigenvectors can be rotated such that at least $m-1$ of them are orthogonal to u [5]. This choice of eigenvectors introduces (at least) $m-1$ zero components in the vector $v=X^Tu$ thus permitting further deflation. Of course, in a numerical context one has to deal with the problem of 'almost vanishing' ν_k's and 'almost equal' eigenvalues. This issues are discussed in [8],[16].

The result of the deflation process is a diagonal matrix $\Lambda'\in\mathbb{R}^{n'\times n'}$, $n'\le n$, which has only simple eigenvalues $\lambda'_j\in\sigma(T)$ and a vector v' whose elements are all nonzero. The eigenvalues of $\Lambda'+\theta\beta_k v'v'^T$ are the eigenvalues of $\tilde{T}$ which are *not* at the same time eigenvalues of T. The eigenvalues of Λ' and $\Lambda'+\theta\beta_k v'v'^T$ *strictly* interlace, i.e. satisfy formulae corresponding to (2.5) but with strict inequality signs.

The deflation process is of great importance for the success of the divide and conquer algorithm. First the investigation of the Weinstein determinant is simplified since its poles coincide with the eigenvalues of Λ' and second n' is often considerably smaller than n [7],[8].

For the computation of the zeros of f a quadratically and monotonically convergent root finder has been proposed by Bunch et al. [5]. As soon as a zero $\tilde{\lambda}$ of f is found, a corresponding eigenvector $\tilde{x}$ of $\tilde{T}$ can be computed by

$$\tilde{x}=(\tilde{\lambda}I-T)^{-1}u=X(\tilde{\lambda}I-\Lambda_1\oplus\tilde{\lambda}I-\Lambda_2)^{-1}v. \tag{2.6}$$

For the complexity analysis for this algorithm we assume that we have no deflation and that in the average s iteration steps are required to obtain one eigenvalue with the mentioned zerofinder. $f(\lambda)$ in (2.4) and its derivative are computed by

$$\begin{aligned} f(\lambda) &= 1+\theta\beta_k v^T h, \qquad h=h(\lambda)=(\Lambda-\lambda I)^{-1}v, \\ f'(\lambda) &= \theta\beta_k h^T h, \end{aligned} \tag{2.7}$$

in $6n+O(1)$ flops. Notice, that after convergence, $h(\tilde{\lambda})$ is an (unnormalized) eigenvector of $\Lambda+\theta\beta_k vv^T$ corresponding to the just found eigenvalue $\tilde{\lambda}$. So, in the average the computation of one eigenvalue together with its normalized eigenvector

costs $(6s+1)n + O(1)$ flops. Due to the special structure of X, the transformation of an eigenvalue of $\Lambda + \theta\beta_k vv^T$ to one of $\tilde{T}$ needs n^2 flops. All together it costs

$$Z(n) = 2Z(\frac{n}{2}) + n^3 + (6s+1)n^2 + O(n) \cong \frac{4}{3}n^3 + 2(6s+1)n^2 + O(n) \tag{2.8}$$

to compute the complete spectral decomposition of $\tilde{T}$, provided that the split eigenvalue problems are solved with the same divide and conquer algorithm.

It is also possible to compute the eigenvectors of $\tilde{T}$ directly by inverse iteration as soon as its eigenvalues are known [10]. In this way the expensive back transformation could be saved and one would get an overall $O(n^2)$ algorithm. Unfortunately, inverse iteration does not yield sets of orthogonal eigenvectors in the presence of close eigenvalues. If the close eigenvalues were not deflated, a further orthogonalization may be necessary, hence bringing back the potentially $O(n^3)$ complexity.

3. Restricted rank modified eigenvalue problems. Let $A \in \mathbb{R}^{n\times n}$ and consider the *modified eigenvalue problem*

$$\tilde{A}x := (A + UV^T)x = \lambda x, \qquad U, V \in \mathbb{R}^{n\times r}, \operatorname{rank}(U) = \operatorname{rank}(V) = r. \tag{3.1}$$

$\tilde{A}$ is called a rank-r modification of A.

We first consider the problem of determining the eigenvalues and eigenvectors of $\tilde{A}$ assuming that we know all the eigenvalues and an orthonormal basis of the corresponding eigenspaces of A. We won't need to know principal vectors of A. The related symmetric eigenvalue problem has been dealt with by several authors [2], [3], [4].

The following propositions are closely related to similar statements for the symmetric eigenvalue problem in Arbenz et al. [2]

PROPOSITION 3.1. *For $\lambda \notin \sigma(A)$, the spectrum of A, the matrices*

$$B := \begin{pmatrix} \lambda - \tilde{A} & 0 \\ 0 & I_r \end{pmatrix} \quad \textit{and} \quad C := \begin{pmatrix} \lambda - A & 0 \\ 0 & Z(\lambda) \end{pmatrix} \tag{3.2}$$

with

$$Z(\lambda) := I_r - V^T(\lambda - A)^{-1}U \tag{3.3}$$

satisfy the equation

$$C = MBN \tag{3.4}$$

where

$$M := \begin{pmatrix} I_n & 0 \\ V^T(A-\lambda)^{-1} & I_r \end{pmatrix}\begin{pmatrix} I_n & U \\ 0 & I_r \end{pmatrix} = \begin{pmatrix} I_n & U \\ V^T(A-\lambda)^{-1} & Z(\lambda) \end{pmatrix} \tag{3.5a}$$

and

$$N := \begin{pmatrix} I_n & 0 \\ V^T & I_r \end{pmatrix}\begin{pmatrix} I_n & (A-\lambda)^{-1}U \\ 0 & I_r \end{pmatrix} = \begin{pmatrix} I_n & (A-\lambda)^{-1}U \\ V^T & Z(\lambda) \end{pmatrix} \tag{3.5b}$$

Proof. By verification. □

Because $\det M = \det N = 1$ for all λ, we have

$$\det(\lambda - \tilde{A}) = \det(\lambda - A) \det Z(\lambda), \qquad \lambda \notin \sigma(A). \tag{3.6}$$

Thus $\lambda \notin \sigma(A)$ is an eigenvalue of $\tilde{A}$ if and only if $\det Z(\lambda) = 0$. For $r = 1$ this becomes the secular equation

$$Z(\lambda) = 1 + v^T(T - \lambda)^{-1}u = 1 + \hat{v}^T(\Lambda - \lambda)^{-1}\hat{u} = 0.$$

Furthermore, as M and N are invertible, the mappings

$$N|_{\mathcal{N}(C)} : \mathcal{N}(C) \longrightarrow \mathcal{N}(B) : x \mapsto Nx \tag{3.7a}$$

and

$$M^H|_{\mathcal{N}(C^H)} : \mathcal{N}(C^H) \longrightarrow \mathcal{N}(B^H) : y \mapsto M^H y \tag{3.7b}$$

are bijective. Because λ is not an eigenvalue of A, x in (3.7a) must be of the form

$$x = \begin{pmatrix} 0 \\ x_2 \end{pmatrix}, \qquad x_2 \in \mathbb{C}^r.$$

Therefore,

$$Nx = \begin{pmatrix} (A - \lambda)^{-1}Ux_2 \\ Z(\lambda)x_2 \end{pmatrix} = \begin{pmatrix} (A - \lambda)^{-1}Ux_2 \\ 0 \end{pmatrix}.$$

By consequence, the mappings

$$(A - \lambda)^{-1}U : \mathcal{N}(Z(\lambda)) \longrightarrow \mathcal{N}(\tilde{A} - \lambda) : z \mapsto (A - \lambda)^{-1}Uz \tag{3.8a}$$

and

$$(A^H - \bar{\lambda})^{-1}V : \mathcal{N}(Z^H(\lambda)) \longrightarrow \mathcal{N}(\tilde{A}^H - \bar{\lambda}) : z \mapsto (A^H - \bar{\lambda})^{-1}Vz \tag{3.8b}$$

are bijective. If $\det Z(\lambda)$ vanishes, we can by (3.8a) compute a basis of the nullspace of $\tilde{A} - \lambda I$ which is the right eigenspace of $\tilde{A}$ corresponding to the eigenvalue λ as soon as we know a basis of $\mathcal{N}(Z(\lambda))$. Likewise, using (3.8b) we can compute a basis of the nullspace of $\tilde{A}^H - \bar{\lambda}I$ which is the left eigenspace of $\tilde{A}$ corresponding to the eigenvalue λ as soon as we know a basis of $\mathcal{N}(Z(\lambda)^H)$.

Note that the *geometric* multiplicity of an eigenvalue of $\tilde{A}$ which is not at the same time an eigenvalue of A cannot exceed r, the order of $Z(\lambda)$. However, the *algebraic* multiplicity can, as is shown by the following example.

Example 3.1. Choose

$$A = \begin{pmatrix} 0 & & & 1 \\ 1 & \ddots & & \\ & \ddots & \ddots & \\ & & 1 & 0 \end{pmatrix} \quad \text{and} \quad \tilde{A} = \begin{pmatrix} 0 & & & \\ 1 & \ddots & & \\ & \ddots & \ddots & \\ & & 1 & 0 \end{pmatrix}$$

Then, $\sigma(A) = \{e^{2\pi ij/n}, j = 0, \ldots, n-1\}$ and $\sigma(\tilde{A}) = \{0\}$. Equation (3.1) holds with

$$U = e_1 = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \qquad V = -e_n = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ -1 \end{pmatrix}.$$

The determinant of the 1×1 Weinstein matrix is

$$Z(\lambda) = \frac{\det(\lambda - \tilde{A})}{\det(\lambda - A)} = \frac{\lambda^n}{1 - \lambda^n}.$$

$\lambda = 0$ is an (algebraically) n-fold eigenvalue of $\tilde{A}$. Its one-dimensional right eigenspace is spanned by (cf. (3.8a))

$$(A - 0)^{-1}U = A^T U = e_n$$

and its left eigenspace by (cf. (3.8b))

$$(A^H - 0)^{-1}V = AV = e_1 .\square$$

To also get right principal vectors corresponding to λ we can proceed in the following bootstrapping manner: Let us define $\mathcal{N}_j := \mathcal{N}((\tilde{A} - \lambda)^j)$. Now we construct recursively bases $\{q_1^{(j)}, \ldots, q_{s_j}^{(j)}\}$ for $\mathcal{N}_j \ominus \mathcal{N}_{j-1} = \mathcal{N}_j \cap \mathcal{N}_{j-1}^{\perp}$ such that $\mathcal{N}_k$ is spanned by $\{q_i^{(j)} | \, 1 \le i \le s_j, 1 \le j \le k\}$, $k = 1, \ldots, q$, where q is the smallest integer for which $\mathcal{N}_{q+1} = \mathcal{N}_q$. By (3.8a) we have already a means to construct a basis of $\mathcal{N}_1$. Notice that $s_1 = \dim \mathcal{N}_1 = \dim \mathcal{N}(Z(\lambda))$ and that $\sum_{j=1}^{q} s_j$ equals the algebraic multiplicity of the eigenvalue λ of $\tilde{A}$.

If $x \in \mathcal{N}_{j+1} \ominus \mathcal{N}_j$, $j \ge 1$, then $y := (\tilde{A} - \lambda)x \in \mathcal{N}_j \ominus \mathcal{N}_{j-1}$. Clearly, $y \in \mathcal{R}(\tilde{A} - \lambda)$, too. If, on the other hand, $y \in (\mathcal{N}_j \ominus \mathcal{N}_{j-1}) \cap \mathcal{R}(\tilde{A} - \lambda)$, then $(\tilde{A} - \lambda)^+ y \in \mathcal{N}_{j+1} \ominus \mathcal{N}_j$. In fact, $(\tilde{A} - \lambda)(\tilde{A} - \lambda)^+$ is the orthogonal projector on $\mathcal{R}(\tilde{A} - \lambda)$ [13, p.423]. So, $(\tilde{A} - \lambda)(\tilde{A} - \lambda)^+ y = y$. Therefore,

$$(\tilde{A} - \lambda)^+ : (\mathcal{N}_j \ominus \mathcal{N}_{j-1}) \cap \mathcal{R}(\tilde{A} - \lambda) \longrightarrow \mathcal{N}_{j+1} \ominus \mathcal{N}_j$$

is bijective.

Now, taking a maximal set of linearly independent vectors $z_1, \ldots, z_{s_{j+1}}$ in $(\mathcal{N}_j \ominus \mathcal{N}_{j-1}) \cap \mathcal{R}(\tilde{A} - \lambda)$, we get by defining

$$q_i^{(j+1)} := (\tilde{A} - \lambda)^+ z_i, \quad i = 1, \ldots, s_{j+1}, \tag{3.9}$$

s_{j+1} linearly independent vectors spanning $\mathcal{N}_j \ominus \mathcal{N}_{j-1}$.

Because $\mathcal{R}(\tilde{A} - \lambda) = \mathcal{N}(\tilde{A}^H - \bar{\lambda})^{\perp}$, a vector $Q_R^{(j)} w := [q_1^{(j)}, \ldots, q_{s_j}^{(j)}]w \in \mathcal{N}_j \ominus \mathcal{N}_{j-1}$ is also in $\mathcal{R}(\tilde{A} - \lambda)$ if and only of

$$Q_L^{(1)H} Q_R^{(j)} w = 0. \tag{3.10}$$

Here, the columns of $Q_L^{(1)} \in \mathbb{C}^{n \times s_1}$ form a basis of $\mathcal{N}(\tilde{A}^H - \bar{\lambda})$.

Using (3.4) in the form $B^+ = NC^+M$, we can reformulate (3.9) as

$$q_i^{(j+1)} = (\lambda - A)^{-1}(I + UZ(\lambda)^+V^T(\lambda - A)^{-1})z_i. \tag{3.11}$$

With (3.10) we see that $z_i = Q_R^{(j)}w_i$ for some $w_i \in \mathcal{N}({Q_L^{(1)}}^H Q_R^{(j)})$. So, we obtain the bijection

$$(A - \lambda)^{-1}(I - UZ(\lambda)^+V^T(A - \lambda)^{-1})Q_R^{(j)} : \mathcal{N}({Q_L^{(1)}}^H Q_R^{(j)}) \longrightarrow \mathcal{N}_{j+1} \ominus \mathcal{N}_j \tag{3.12}$$

which enables us to compute the principal vectors of grade $j > 1$.

Example 3.1. [Continued] With the above we can set

$$Q_R^{(1)} = e_n\,,\ Q_L^{(1)} = e_1\,.$$

As ${Q_L^{(1)}}^H Q_R^{(1)} = 0$, there is a right principal vector of grade 2, obtainable by (3.12)

$$Q_R^{(2)} = (A - \lambda)^{-1}(I - e_1 0 e_n^T(A - \lambda)^{-1})e_n = A^T e_n = e_{n-1}\,.$$

We continue as described until we get $Q_R^{(n)} = e_1$. At this point

$${Q_L^{(1)}}^H Q_R^{(n)} = 1 \neq 0$$

and the process breaks down. □

Left principal vectors can be obtained in a similar way, starting with the columns of $Q_L^{(1)}$ which span $\mathcal{N}(\bar{A}^H - \bar{\lambda})$.

Proposition 3.1 can be generalized to the case where λ is an eigenvalue of the unmodified matrix A.

PROPOSITION 3.2. *Let λ be an eigenvalue of A with geometric multiplicity m. Let $W_L, W_R \in \mathbb{C}^{n\times m}$ be orthogonal matrices spanning $\mathcal{N}(A^H - \bar{\lambda})$ and $\mathcal{N}(A - \lambda)$, respectively. Then the matrices*

$$B = \begin{pmatrix} \lambda - A - UV^T & 0 & 0 \\ 0 & I_r & 0 \\ 0 & 0 & 0_m \end{pmatrix} \tag{3.13a}$$

and

$$C = \begin{pmatrix} \lambda - A & 0 & 0 \\ 0 & I_r - V^T(\lambda - A)^+U & V^TW_R \\ 0 & W_L^HU & 0_m \end{pmatrix} \tag{3.13b}$$

satisfy the equation

$$C = MBN \tag{3.14}$$

where

$$\begin{aligned} M &= \begin{pmatrix} I_n & 0 & 0 \\ -V^T(\lambda - A)^+ & I_r & 0 \\ 0 & 0 & I_m \end{pmatrix} \begin{pmatrix} I_n & (I - W_LW_L^H)U & 0 \\ 0 & I_r & 0 \\ 0 & W_L^HU & I_m \end{pmatrix} \\ &\qquad \begin{pmatrix} I_n - W_LW_L^H & 0 & W_L \\ 0 & I_r & 0 \\ W_L^H & 0 & 0 \end{pmatrix} \qquad (3.15a) \\ &= \begin{pmatrix} I_n - W_LW_L^H & (I - W_LW_L^H)U & W_L \\ -V^T(\lambda - A)^+ & I_r - V^T(\lambda - A)^+U & 0 \\ W_L^H & W_L^HU & 0 \end{pmatrix} \end{aligned}$$

and, similarly,

$$N = \begin{pmatrix} I_n - W_R W_R^H & -(\lambda - A)^+ U & W_R \\ V^T(I_n - W_R W_R^H) & I_r - V^T(\lambda - A)^+ U & V^T W_R \\ W_R^H & 0 & 0 \end{pmatrix}. \tag{3.15b}$$

Proof. By Verification. □

The submatrix

$$Z_e(\lambda) = \begin{pmatrix} I_r - V^T(\lambda - A)^+ U & V^T W_R \\ W_L^H U & 0 \end{pmatrix} \in \mathbb{C}^{(r+m)\times(r+m)} \tag{3.16}$$

of C is a generalization of $Z(\lambda)$ in (3.3). $(\lambda - A)^+$ is the Moore–Penrose generalized inverse of $\lambda - A$. In the real symmetric case $Z(\lambda)$ and $Z_e(\lambda)$ are called Weinstein and extended Weinstein matrix [19].

Let x be a right eigenvector of A corresponding to λ. Then $\bar{x} := \left(x^T, 0^T, 0^T\right)^T \in \mathbb{C}^{n+r+m}$ is in the nullspace of C in (3.13b) and, as N is nonsingular, $z := N\bar{x}$ is in the nullspace of B. But

$$z := N\bar{x} = N \begin{pmatrix} x \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ W_R^H x \end{pmatrix}.$$

So, z is an eigenvector corresponding to the zero diagonal block of B. If $Z_e(\lambda)$ is singular and $\begin{pmatrix} w_1 \\ w_2 \end{pmatrix} \in \mathcal{N}(Z_e(\lambda))$, then

$$z := N \begin{pmatrix} 0 \\ w_1 \\ w_2 \end{pmatrix} = \begin{pmatrix} x_1 \\ 0 \\ 0 \end{pmatrix}.$$

where

$$x_1 = -(\lambda - A)^+ U w_1 + W_R w_2 \tag{3.17a}$$

is an eigenvector of $A + UV^T$ corresponding to λ. Thus (3.17a) is a bijective map from $\mathcal{N}(Z_e(\lambda))$ onto $\mathcal{N}(\tilde{A} - \lambda)$. In a similar way M^H yields a mapping from $\mathcal{N}(Z_e^H(\lambda))$ onto $\mathcal{N}(\tilde{A}^H - \bar{\lambda})$:

$$y_1 = -(\bar{\lambda} - A^H)^+ V u_1 + W_L u_2, \qquad \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} \in \mathcal{N}(Z_e^H(\lambda)). \tag{3.17b}$$

To obtain principal vectors for $\lambda \in \sigma(\tilde{A}) \cap \sigma(A)$ we can proceed in a similar way as for $\lambda \notin \sigma(A)$. We have to replace however formula (3.11) by

$$q_i^{(j+1)} = (\lambda - A)^+ z_i + [(\lambda - A)^+ U, W_R] Z_e(\lambda)^+ \begin{bmatrix} V^T(\lambda - A)^+ \\ W_L \end{bmatrix} z_i. \tag{3.18}$$

This can be derived from (3.14). Again, $z_i = Q_R^{(j)} w_i$ for some $w_i \in \mathcal{N}({Q_L^{(1)}}^H Q_R^{(j)})$. Therefore,

$$(\lambda - A)^+ + [(\lambda - A)^+ U, W_R] Z_e(\lambda)^+ \begin{bmatrix} V^T(\lambda - A)^+ \\ W_L \end{bmatrix} : \mathcal{N}({Q_L^{(1)}}^H Q_R^{(j)}) \longrightarrow \mathcal{N}_{j+1} \ominus \mathcal{N}_j$$

is bijective.

Proceeding in the above way, we get for every $\lambda \in \sigma(\tilde{A})$ a set of eigen and principal vectors $\{q_1^{(1)}, \ldots, q_{s_q}^{(q)}\}$. The integer q as well as the s_i are of course dependent on λ. By construction, we have $(\tilde{A} - \lambda)q_i^{(j)} = \sum_{l=1}^{s_{j-1}} \eta_{il} q_l^{(j-1)}$. Therefore,

$$\tilde{A}\left[q_1^{(1)}, \ldots, q_{s_1}^{(1)}, q_1^{(2)}, \ldots, q_{s_q}^{(q)}\right] = \left[q_1^{(1)}, \ldots, q_{s_1}^{(1)}, q_1^{(2)}, \ldots, q_{s_q}^{(q)}\right] \begin{pmatrix} \lambda I_{s_1} & E_{s_1 \times s_2} & & & \\ & \lambda I_{s_2} & E_{s_2 \times s_3} & & \\ & & \ddots & \ddots & \\ & & & \ddots & E_{s_{q-1} \times s_q} \\ & & & & \lambda I_{s_q} \end{pmatrix}. \tag{3.19}$$

Defining by $\tilde{Q}$ the matrix containing the eigen and principal vectors of *all* eigenvalues of $\tilde{A}$ such that vectors corresponding to the same eigenvalue appear in blocks as in (3.19), we get

$$\tilde{A}\tilde{Q} = \tilde{Q}\tilde{\Delta} \tag{3.20}$$

where $\tilde{\Delta}$ is block diagonal with block upper bidiagonal diagonal blocks.

From (3.19) we see that $\tilde{\Delta}$ is upper triangular. Let $\tilde{Q} = \tilde{X}\tilde{R}$ be the QR decomposition of $\tilde{Q}$. Then

$$\tilde{A}\tilde{X} = \tilde{X}(R\tilde{\Delta}R^{-1}) \tag{3.21}$$

is a Schur normal form of $\tilde{A}$. $R\tilde{\Delta}R^{-1}$ is upper triangular, having the eigenvalues arranged in its diagonal as in $\tilde{\Delta}$.

4. Deflation. In this section we show how the extended matrix $Z_e(\lambda)$ can be transformed to a simpler matrix, thereby making the calculation of some or all of the left and right null spaces of $\tilde{A}$ associated with λ an easier task.

4.1. The rank-r case. In order to calculate x_1 and y_1 in (3.17), we need to calculate the null spaces of $Z_e(\lambda)$ and its conjugate transpose from (3.16). When $m > r$ we can *deflate* the problem by pre and post multiplication of $Z_e(\lambda)$ to get a simpler matrix $\hat{Z}_e(\lambda)$,

$$\hat{Z}_e(\lambda) = \begin{pmatrix} I_r & 0 \\ 0 & Q_L \end{pmatrix} Z_e(\lambda) \begin{pmatrix} I_r & 0 \\ 0 & Q_R^H \end{pmatrix}, \quad Q_L, Q_R \in \mathbb{C}^{m \times m} \tag{4.1}$$

If $\begin{pmatrix} \hat{w}_1 \\ \hat{w}_2 \end{pmatrix} \in \mathcal{N}(\hat{Z}_e(\lambda))$ and $\begin{pmatrix} \hat{u}_1 \\ \hat{u}_2 \end{pmatrix} \in \mathcal{N}(\hat{Z}_e{}^H(\lambda))$ then $\begin{pmatrix} \hat{w}_1 \\ Q_R^H \hat{w}_2 \end{pmatrix} \in \mathcal{N}(Z_e(\lambda))$ and $\begin{pmatrix} \hat{u}_1 \\ Q_L^H \hat{u}_2 \end{pmatrix} \in \mathcal{N}(Z_e^H(\lambda))$, respectively. From (3.17) we see that

$$x_1 = -(\lambda I - A)^+ U \hat{w}_1 + W_R Q_R^H \hat{w}_2 \tag{4.2a}$$

is a right eigenvector of $\tilde{A}$ corresponding to λ and that

$$y_1 = -(\bar{\lambda} I - A^H)^+ V \hat{u}_1 + W_L Q_L^H \hat{u}_2 \tag{4.2b}$$

satisfies $y_1^H(\tilde{A} - \lambda I) = 0$. Hence we call y_1 a left eigenvector of $\tilde{A}$ corresponding to λ. Thus the bases used for $\mathcal{N}(A - \lambda I)$ and $\mathcal{N}(A^H - \bar{\lambda} I)$ have been changed from W_R to $W_R Q_R^H$ and from W_L to $W_L Q_L^H$, respectively.

To find $\hat{Z}_e(\lambda)$, we choose Q_L to be a sequence of Householder reflectors so that $Q_L W_L^H U = \begin{pmatrix} \Pi_r \\ 0 \end{pmatrix}$ where Π_r is an upper triangular $r \times r$ matrix. Likewise, Q_R^H is chosen so that $V^T W_R Q_R^H = (\Phi_r, 0)$ where Φ_r is an $r \times r$ lower triangular matrix. The new matrix has the form

$$\hat{Z}_e(\lambda) = \begin{pmatrix} \Sigma_r & \Phi_r & 0 \\ \Pi_r & 0_r & 0 \\ 0 & 0 & 0_{m-r} \end{pmatrix} \tag{4.3}$$

where $\Sigma_r = I_r - V^T(\lambda I - A)^+ U$ and 0_k denotes a square $k \times k$ zero matrix. We also denote the leading 2×2 block principal submatrix of (4.3) as $\hat{Z}_{e,11}(\lambda)$.

Let $\dim \mathcal{N}(\hat{Z}_{e,11}) = \mu$ and $\hat{W} \in \mathbb{C}^{2r \times \mu}$ be a matrix, the columns of which form a basis of $\mathcal{N}(\hat{Z}_{e,11})$. Then

$$\mathcal{R}\begin{pmatrix} 0_{2r \times (m-r)} & \hat{W} \\ I_{m-r} & 0_{(m-r) \times \mu} \end{pmatrix} = \mathcal{N}(\hat{Z}_e(\lambda)).$$

From (4.2a), the columns of the $n \times (m - r - \mu)$ matrix

$$\begin{pmatrix} (A - \lambda)^+ U & W_R Q_R^H \end{pmatrix} \begin{pmatrix} 0_{2r \times (m-r)} & \hat{W}_{2r \times \mu} \\ I_{m-r} & 0_{(m-r) \times \mu} \end{pmatrix} \tag{4.4a}$$

form a basis of $\mathcal{N}(\tilde{A} - \lambda)$. Likewise, from (4.2b), the columns of the $n \times (m - r + \mu)$ matrix

$$\begin{pmatrix} (A^H - \bar{\lambda})^+ V & W_L Q_L^H \end{pmatrix} \begin{pmatrix} 0_{2r \times (m-r)} & \hat{U}_{2r \times \mu} \\ I_{m-r} & 0_{(m-r) \times \mu} \end{pmatrix} \tag{4.4b}$$

form a basis of $\mathcal{N}(\tilde{A}^H - \bar{\lambda})$. Here, $\mathcal{R}(\hat{U}) = \mathcal{N}(\hat{Z}_{e,11}^H)$.

If $m \leq r$, we choose $Q_L = Q_R = I$. This means $\hat{Z}_e(\lambda) = Z_e(\lambda)$, where

$$Z_e(\lambda) = \begin{pmatrix} \Sigma_{r \times r} & \Phi_{r \times m} \\ \Pi_{m \times r} & 0_{m \times m} \end{pmatrix}. \tag{4.5}$$

For this case, the original equations in (3.17) apply. Φ or Π may be rank deficient. We do not elaborate this situation in this paper, except for the most interesting special case $r = 1$ of the following subsection.

4.2. The rank-1 case. We now show how the results in (4.4a) and (4.4b) are simplified for the case $m > r$ when U and V are $n \times 1$ vectors; that is, $\tilde{A}$ is gotten from A by a rank-1 update. This is an important case, since both nonsymmetric tridiagonal and Hessenberg matrices can be decoupled in this fashion, (see Section 6).

From (4.4a) and (4.4b) we see that the last $m - 1$ columns of $W_R Q_R^H$ and $W_L Q_L^H$ are right and left eigenvectors of $\tilde{A}$ corresponding to λ, respectively. To see if there

are any more right and left eigenvectors, we must examine the null spaces of $\hat{Z}_{e,11}(\lambda)$ and its conjugate transpose, where

$$\hat{Z}_{e,11}(\lambda) = \begin{pmatrix} \sigma & \phi \\ \pi & 0 \end{pmatrix}. \tag{4.6}$$

We have five cases to consider.

1. If $\pi\phi \neq 0$ then rank $\hat{Z}_{e,11}(\lambda) = 2$ and we get no more right or left eigenvectors, and $\dim\mathcal{N}(\tilde{A} - \lambda I) = m - 1$.
2. If $\phi = 0, \pi \neq 0$ then rank $\hat{Z}_{e,11}(\lambda) = 1$. In this case, $(0\ 1)^T$ spans $\mathcal{N}(\hat{Z}_{e,11}(\lambda))$ and $x_1 = W_R Q_R^H (1\ 0)^T$, the first column of $W_R Q_R^H$, is the m-th right eigenvector and $\dim\mathcal{N}(\tilde{A} - \lambda I) = m$. Likewise, $y_1 = (A^H - \bar{\lambda} I)^+ V + W_L Q_L^H (-\frac{\sigma}{\pi}\ 0)^T$ is the m-th left eigenvector.
3. If $\phi \neq 0$ and $\pi = 0$, then the rank of $\hat{Z}_{e,11}(\lambda)$ is 1. $x_1 = (A - \lambda I)^+ U + W_R Q_R^H (-\frac{\sigma}{\phi}\ 0)^T$ is the m-th right eigenvector, and $\dim\mathcal{N}(\tilde{A} - \lambda I) = m$. Likewise, $y_1 = W_L Q_L^H (1\ 0)^T$ is the m-th left eigenvector.
4. If $\phi = \pi = 0$, $\sigma \neq 0$, $\hat{Z}_{e,11}(\lambda)$ has rank 1. We easily find that $x_1 = W_R Q_R^H (1\ 0)^T$ and $y_1 = W_L Q_L^H (1\ 0)^T$ are the m-th right and left eigenvectors of $\tilde{A}$, respectively, and $\dim\mathcal{N}(\tilde{A} - \lambda I) = m$.
5. If $\phi = \pi = \sigma = 0$, $\hat{Z}_{e,11}$ equals the 2×2 zero matrix and has a null space spanned by e_1 and e_2. Hence the m-th eigenvectors are given as in case 4. The $m+1$-st right and left eigenvectors are given by $x_1 = (A - \lambda I)^+ U$ and $y_1 = (A^H - \bar{\lambda} I)^+ V$, respectively, and $\dim\mathcal{N}(\tilde{A} - \lambda I) = m + 1$.

In the case $m = r = 1$, the original matrix $Z_e(\lambda)$ is

$$Z_e(\lambda) = \begin{pmatrix} \sigma & \phi \\ \pi & 0 \end{pmatrix}. \tag{4.7}$$

This matrix is singular if and only if $\pi\phi = 0$. Hence, the left and right eigenvectors are given in cases 2-5 with $Q_L = Q_R = I$.

After having determined left and right eigenspaces and computed the extended Weinstein matrix for all $\lambda \in \sigma(A)$ one may wonder if this information can be used to reduce the cost of the search of the remaining eigenvalues $\lambda \in \sigma(\tilde{A}) \setminus \sigma(A)$. We know that left and right eigenvectors corresponding to different eigenvalues are orthogonal, but we see no way to reduce the order of the given matrix eigenvalue problem for the computation of the remaining eigenvalues as can be done in the symmetric eigenvalue problem in the deflation process. In order to do so, we have – as in the symmetric eigenvalue problem – to assume that we have more information available, more precisely that we know the spectral decomposition of A.

If A is diagonalizable, $AX = X\Lambda$, $\Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_n)$, we can proceed similarly as in the symmetric tridiagonal eigenvalue problem. We can deflate in the proper meaning of the word, i.e. we can transform the original modified eigenvalue problem into a modified eigenvalue problem of lower order, containing all the information to compute the *new* eigenvalues not in $\sigma(A)$ and corresponding eigenvectors. Let $X = [x_1, \ldots, x_n]$ and $Y := (X^{-1})^H = [y_1, \ldots, y_n]$. Then

$$A = X\Lambda Y^H = \sum_{i=1}^{n} \lambda_i x_i y_i^H. \tag{4.8}$$

Bases for left and right eigenspaces corresponding to an m-fold eigenvalue λ_k, say, are obtained by selecting those vectors x_{i_j} and y_{i_j} for which $\lambda_{i_j} = \lambda_k, j = 1, \ldots, m$. Notice, that diagonalizable matrices have no degenerate eigenvalues.

Using the spectral decomposition (4.8), the modified matrix $\tilde{A}$ becomes

$$\tilde{A} = X(\Lambda + \hat{u}\hat{v}^H)Y^H, \qquad \hat{u} = Y^H u, \quad \hat{v} = X^H v. \tag{4.9}$$

If $\Lambda + \hat{u}\hat{v}^H$ has the spectral decomposition $\Lambda + \hat{u}\hat{v}^H = \hat{X}\tilde{\Lambda}\hat{Y}^H$, then

$$\tilde{A} = \tilde{X}\tilde{\Lambda}\tilde{Y}^H, \qquad \tilde{X} = X\hat{X}, \quad \tilde{Y} = Y\hat{Y}. \tag{4.10}$$

So we have to investigate the eigenvalue problem

$$(\Lambda + \hat{u}\hat{v}^H)y = \lambda y \tag{4.11}$$

to get the spectral decomposition of $\tilde{A}$. As left and right eigenvectors of Λ corresponding to the above eigenvalue λ_k are given by the unit vectors $e_{i_1}, \ldots, e_{i_m}$, the extended Weinstein matrix in λ_k becomes

$$Z_e(\lambda_k) = \begin{pmatrix} 1 - \sum_{\lambda_i \neq \lambda_k} \frac{\bar{\hat{v}}_i \hat{u}_i}{\lambda_k - \lambda_i} & \bar{\hat{v}}_{i_1} & \cdots & \bar{\hat{v}}_{i_m} \\ \hat{u}_{i_1} & & & \\ \vdots & & & \\ \hat{u}_{i_m} & & & \end{pmatrix}.$$

We now chose $Q = Q(\lambda_k) \in \mathbb{C}^{m \times m}$ unitary such that

$$Q^H \begin{pmatrix} \hat{v}_{i_1} \\ \vdots \\ \hat{v}_{i_m} \end{pmatrix} = \overline{\phi(\lambda_k)} e_1, \qquad |\phi(\lambda_k)|^2 = \sum_{j=1}^{m} |\hat{v}_{i_j}|^2,$$

and replace columns $i_1, \ldots, i_m$ in X and Y by

$$[x_{i_1}, \ldots, x_{i_m}]Q, \qquad [y_{i_1}, \ldots, y_{i_m}]Q.$$

As we transformed left and right eigenvectors by the same Q, (4.8) and (4.9) still hold with the new X and Y, but now $v_{i_j} = 0$, $j = 2, \ldots, m$. So, $x_{i_2}, \ldots, x_{i_m}$ are eigenvectors of $\tilde{A}$ corresponding to the eigenvalue λ_k. If $\phi(\lambda_k) = 0$, x_{i_1} is a further eigenvector corresponding to this same eigenvalue λ_k.

We can proceed in this way for every eigenvalue of A and finally permute the columns of X and Y such that $\hat{v}$ in (4.9) has n' leading nonzero elements followed by $n - n'$ zeros:

$$\tilde{A} = [X_1, X_2] \begin{pmatrix} \Lambda_1 + \hat{u}\hat{v}^H & 0 \\ 0 & \Lambda_2 \end{pmatrix} [Y_1, Y_2]^H, \quad X_1, Y_1 \in \mathbb{C}^{n \times n'}, X_2, Y_2 \in \mathbb{C}^{n \times (n-n')}. \tag{4.12}$$

Note, that Λ_1 has only simple eigenvalues. We can now restrict ourselves to the modified eigenvalue problem (4.11) where $\hat{v}$ has no vanishing element. The vector $\hat{u}$ may have zero components however. If $\hat{u}_i = 0$, then λ_i, the i-th diagonal element of Λ_1 is an eigenvalue of $\Lambda + \hat{u}\hat{v}^H$. This follows immediately from the fact that e_i is a

left eigenvector of this matrix. A right eigenvector corresponding to λ_i is obtained by applying the third of the above 5 cases: If $\begin{pmatrix} \omega_1 \\ \omega_2 \end{pmatrix}$, $|\omega_1|^2 + |\omega_1|^2 \neq 0$, satisfies

$$\begin{aligned} \hat{Z}_e(\lambda_i) \begin{pmatrix} \omega_1 \\ \omega_2 \end{pmatrix} &= \begin{pmatrix} 1 + \hat{v}^h (A - \lambda_i)^+ \hat{u} & \hat{v}_{i_1} \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \omega_1 \\ \omega_2 \end{pmatrix} \\ &= \begin{pmatrix} 1 + \sum_{j \neq i} \frac{\bar{\hat{v}}_i \hat{u}_i}{\lambda_j - \lambda_i} & \hat{v}_{i_1} \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \omega_1 \\ \omega_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \end{aligned}$$

then

$$(A - \lambda_i)^+ - \omega_2/\omega_1 e_i = \sum_{j \neq i} \frac{\hat{u}_j}{\lambda_j - \lambda_i} x_j - \omega_2/\omega_1 e_i \tag{4.13}$$

is a right eigenvector corresponding to λ_i. Notice that we have to form the upper-left element of $\hat{Z}_e(\lambda_i)$ only in the practically rare situation when $\hat{u}_i = 0$.

We cannot deflate (4.11) any further by replacing one of X_1's columns by the vector in (4.13), as the resulting X would not satisfy (4.8) and (4.9) anymore.

The secular equation for (4.12) is

$$Z(\lambda) = 1 - \sum_{i=1}^{n'} \frac{\bar{\hat{v}}_i \hat{u}_i}{\lambda - \lambda_i} = 0. \tag{4.14}$$

We search the zeros of $Z(\lambda) = 0$ to obtain the eigenvalues of $\tilde{A}$ which are not in the spectrum of A. If $Z(\lambda) = 0$, by (3.8a)

$$(\Lambda_1 - \lambda)^{-1} \hat{u}$$

is a corresponding eigenvector of $\Lambda_1 + \hat{u}\hat{v}^H$.

It is possible that $Z(\lambda)$ vanishes in a point $\tilde{\lambda} \in \sigma(A)$. If $\tilde{\lambda} = \lambda_i \in \sigma(\Lambda_1)$ for some $i \leq n'$, then $\hat{u}_i = 0$ and

$$\hat{Z}_e(\tilde{\lambda}) = \begin{pmatrix} 0 & 0 \\ \hat{v}_i & 0 \end{pmatrix}$$

In this situation we choose $\omega_1 = 1$, and $\omega_2 = 0$ in (4.13). If $\tilde{\lambda} \in \sigma(A) \setminus \sigma(\Lambda_1)$, then simply $\dim \mathcal{N}(\tilde{A} - \tilde{\lambda}) = \dim \mathcal{N}(A - \tilde{\lambda}) + 1$.

If we are able to find n' linearly independent eigenvectors $y_1, \ldots, y_{n'}$ of $\Lambda_1 + \hat{u}\hat{v}^H$ in (4.12), $\tilde{A}$ can be diagonalized by means of

$$\tilde{X} = [X_1, X_2](Y \oplus I_{n-n'}), \qquad Y = [y_1, \ldots, y_{n'}].$$

If A is *not* diagonalizable, we still can proceed as described for the nondefective eigenvalues. If the eigenvalue under consideration, say λ, is *defective*, we can arrange the corresponding eigen and principal vectors according to (3.19). The transformation of $Z_e(\lambda)$ to $\hat{Z}_e(\lambda)$ can be performed as before. Here, however, we have principal vectors depending on certain eigenvectors. It may be possible to further rotate the potentially deflatable vectors $x_{i_2}, \ldots, x_{i_m}$ (and eventually x_{i_1}) such that zero rows are introduced into the block $E_{s_1 \times s_2}$ at position $(1,2)$ in the small matrix on the right-hand side of (3.19). Vectors corresponding to such rows can be deflated as no principal vector depends on them. It may also be possible to deflate a whole chain of eigenvectors and principal vectors. Checking this would be an involved procedure. As we doubt its practical importance, we do not want to elaborate it further.

5. Alternate Theory. Let us now assume that A has q eigenvalues $\lambda_1, \ldots, \lambda_q$ with algebraic multiplicities $\mu_1, \ldots, \mu_q$ and that we know a decomposition of A of the form

$$AX = X\Delta, \qquad \Delta = \oplus_{j=1}^{q} \Delta_j, \quad \Delta_j \in \mathbb{C}^{\mu_j \times \mu_j}, \tag{5.1}$$

where Δ is a block-diagonal matrix with blocks $\Delta_1, \ldots, \Delta_q$. The block Δ_j corresponds to the eigenvalue λ_j. In section 3 we computed the eigenvectors and principal vectors of the modified matrix $\tilde{A}$ in a way that a decomposition of the form (5.1) resulted. In [9], Fletcher and Sorensen have shown how such a decomposition can be computed from the Schur normal form. Let's split $X := [X_1, \ldots, X_q]$ corresponding to Δ, $X_j \in \mathbb{C}^{n \times \mu_j}$. Define $Y := (X^{-1})^H = [Y_1, \ldots, Y_q]$, $Y_j \in \mathbb{C}^{n \times \mu_j}$. Then from $Y^H X = XY^H = I_n$ one immediately sees, that

$$Y_i^H X_j = \begin{cases} 0_{\mu_i \times \mu_j}, & i \neq j \\ I_{\mu_j}, & i = j \end{cases}; \qquad I = \sum_{j=1}^{q} X_j Y_j^H. \tag{5.2}$$

Furthermore,

$$A = X\Delta Y^H = \sum_{j=1}^{q} X_j \Delta_j Y_j^H. \tag{5.3}$$

When applying Proposition 3.1 we can make use of representation (5.3). The Weinstein matrix $Z(\lambda)$ of (3.3) then becomes

$$Z(\lambda) = I_r - V^T(\lambda I - A)^{-1}U = I_r - \hat{V}^H(\lambda I - \Delta)^{-1}\hat{U} \tag{5.4}$$

where $\hat{V} = X^H V$, $\hat{U} = Y^H U$. This representation is advantageous if $Z(\lambda)$ has to be formed for many values of λ. In the symmetric case, A is diagonalizable and Δ therefore is diagonal. The cost of forming $Z(\lambda)$ with given $\hat{V} = \hat{U}$ is then only $(2r+1)rn + O(n)$ flops. Working with the original form yields in general a complexity of $O(rn^2 + r^2 n)$ flops. In the nonsymmetric case, A will in most situations have $q \approx n$ eigenvalues which results in small diagonal blocks Δ_j and $O(r^2 n)$ complexity. But large diagonal blocks are possible. An extreme example with $q = 1$ is given by Example 3.1. Using decomposition (5.3) we can formulate a statement similar to Proposition 3.2,

PROPOSITION 5.1. *Let λ_k be an eigenvalue of A with algebraic multiplicity μ_k and let λ be in a neighborhood $N_\epsilon(\lambda_k)$ of λ_k with $N_\epsilon(\lambda_k) \cap \sigma(A) = \{\lambda_k\}$. With the decomposition* (5.1) *the relation*

$$C = N^H B N \tag{5.5}$$

holds where

$$B = \begin{pmatrix} \lambda - A - UV^T & 0 & 0 \\ 0 & I_r & 0 \\ 0 & 0 & \lambda - \Delta_k \end{pmatrix}, \tag{5.6a}$$

$$C = \begin{pmatrix} \lambda - A & 0 & 0 \\ 0 & I_r - V^T(\sum_{\substack{j=1 \\ j \neq k}}^{q} X_j(\lambda - \Delta_j)^{-1} Y_j^H)U & V^T X_k \\ 0 & Y_k^H U & \lambda - \Delta_k \end{pmatrix}, \tag{5.6b}$$

and

$$
\begin{aligned}
N &= \begin{pmatrix} I_n - X_k Y_k^H & 0 & X_k \\ 0 & I_r & 0 \\ Y_k^H & 0 & 0 \end{pmatrix} \begin{pmatrix} I_n & 0 & 0 \\ V^T(I - X_k Y_k^H) & I_r & V^T X_k \\ 0 & 0 & I_{\mu_k} \end{pmatrix} \\
&\qquad \begin{pmatrix} I_n & -\sum_{\substack{j=1 \\ j\neq k}}^{q} X_j(\lambda - \Delta_j)^{-1} Y_j^H)U & 0 \\ 0 & I_r & 0 \\ 0 & 0 & I_{\mu_k} \end{pmatrix} \qquad (5.7) \\
&= \begin{pmatrix} I_n - X_k Y_k^H & -\sum_{\substack{j=1 \\ j\neq k}}^{q} X_j(\lambda - \Delta_j)^{-1} Y_j^H)U & X_k \\ V^T(I - X_k Y_k^H) & I_r - V^T(\sum_{\substack{j=1 \\ j\neq k}}^{q} X_j(\lambda - \Delta_j)^{-1} Y_j^H)U & V^T X_k \\ Y_k^H & 0 & 0 \end{pmatrix}.
\end{aligned}
$$

Proof. By Verification. □

Taking determinants in (5.5) yields

$$\det \hat{Z}_e(\lambda) = (\lambda - \lambda_k)^{\mu_k} \det(\lambda - \tilde{A}) / \det(\lambda - A). \tag{5.8}$$

where

$$\hat{Z}_e(\lambda) = \begin{pmatrix} I_r - V^T(\sum_{\substack{j=1 \\ j\neq k}}^{q} X_j(\lambda - \Delta_j)^{-1} Y_j^H)U & V^T X_k \\ Y_k^H U & \lambda - \Delta_k \end{pmatrix} \in \mathbb{C}^{r+\mu_k \times r+\mu_k} \tag{5.9}$$

is a submatrix of C. Clearly, $\det \hat{Z}_e(\lambda)$ is an analytic function in $N_\epsilon(\lambda_k)$. Note that $\sum_{j\neq k} X_j(\lambda_k - \Delta_j)^{-1} Y_j^H = (\lambda_k - A)^+$ only if geometric and algebraic multiplicity of λ_k coincide. In this case $\hat{Z}_e(\lambda)$ equals $Z_e(\lambda)$ in (3.16).

As earlier, N maps the nullspace of C on the nullspace of B. If $a \in \mathbb{C}^{\mu_k}$ and $X_k a \in \mathcal{N}(A - \lambda_k)$ then $N \begin{pmatrix} X_k a \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ a \end{pmatrix}$ is in the nullspace of B corresponding to the block $\lambda_k - \Delta_k$. Conversely, a vector $\begin{pmatrix} w_1 \\ w_2 \end{pmatrix} \in \mathcal{N}(\hat{Z}_e(\lambda_k))$ is mapped on

$$x_1 = -\sum_{\substack{j=1 \\ j\neq k}}^{q} X_j(\lambda_k - \Delta_j)^{-1} Y_j^H U w_1 + X_k w_2 \in \mathcal{N}(\tilde{A} - \lambda_k) \tag{5.10}$$

Similarly, a vector $\begin{pmatrix} u_1 \\ u_2 \end{pmatrix} \in \mathcal{N}(Z_e^H(\lambda_k))$ is mapped on $\mathcal{N}(\tilde{A}^H - \bar{\lambda}_k)$ by

$$y_1 = -\sum_{\substack{j=1 \\ j\neq k}}^{q} Y_j(\bar{\lambda}_k - \Delta_j^H)^{-1} X_j^H V u_1 + Y_k u_2 . \tag{5.11}$$

One proceeds as in section 3 to obtain principal vectors. But here one can use representation (5.3) in (3.8), and (3.18).

6. Application of the Theory. In this section, we apply the theory of the last sections to two special cases – the nonsymmetric tridiagonal eigenvalue problem and the Hessenberg eigenvalue problem.

6.1. The tridiagonal eigenvalue problem. We consider now the eigenvalue problem

$$\tilde{T}x = \lambda x, \tag{6.1}$$

where $\tilde{T}$ is the nonsymmetric irreducible tridiagonal matrix

$$\tilde{T} = \begin{pmatrix} \alpha_1 & \beta_1 & & & \\ \gamma_2 & \ddots & \ddots & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & \beta_{n-1} \\ & & & \gamma_n & \alpha_n \end{pmatrix}. \tag{6.2}$$

We split $\tilde{T}$ into T and a rank-1-modification. To zero out the elements of T at position β_k and γ_{k+1} simultaneously we have to choose the splitting vectors u and v appropriately:

$$\begin{aligned} \tilde{T} &= T + \begin{pmatrix} 0 & & & \\ & \frac{1}{\omega}\beta_k & \beta_k & \\ & \gamma_{k+1} & \omega\gamma_{k+1} & \\ & & & 0 \end{pmatrix}, \qquad \omega \neq 0 \\ &= \begin{pmatrix} T_1 & 0 \\ 0 & T_2 \end{pmatrix} + \begin{pmatrix} \frac{1}{\omega}\beta_k e_k \\ \gamma_{k+1} e_1 \end{pmatrix} \left(e_k^T, \omega e_1^T \right) = T + uv^T. \end{aligned} \tag{6.3}$$

In this case, we need to find the eigenvalues, λ, of the block diagonal matrix T and the associated eigenspaces $W_R = \mathcal{N}(T - \lambda I)$ and $W_L = \mathcal{N}(T^H - \bar{\lambda} I)$. Then the extended matrix given in (3.16) with $V^T = (e_k^T \;\; \omega e_1^T)$, $U^T = (\frac{1}{\omega}\beta_k e_k^T \;\; \gamma_{k+1} e_1^T)$, $\tilde{A} = \tilde{T}$, and $A = T$ must be formed. If this matrix has a nontrivial null space then λ is an eigenvalue of $\tilde{T}$ and the respective spaces $\mathcal{N}(\tilde{T} - \lambda I)$ and $\mathcal{N}(\tilde{T}^H - \bar{\lambda} I)$ are computed using the five cases in section 4.2. We will not assume that $\tilde{T}$ is diagonalizable. Hence, we will not take advantage of the savings in cost that could possibly be obtained by a reduction to a smaller problem as described in section 4.2. Next, to find the eigenvalues λ of $\tilde{T}$ that are not eigenvalues of T, we can find the roots of the scalar secular equation

$$\begin{aligned} Z(\lambda) &= 1 + \begin{pmatrix} e_k \\ \omega e_1 \end{pmatrix}^T (T - \lambda)^{-1} \begin{pmatrix} \frac{1}{\omega}\beta_k e_k \\ \gamma_{k+1} e_1 \end{pmatrix} \\ &= 1 + \tfrac{\beta_k}{\omega} e_k^T (T_1 - \lambda)^{-1} e_k + \omega\gamma_{k+1} e_1^T (T_2 - \lambda)^{-1} e_1 = 0. \end{aligned} \tag{6.4}$$

Then the associated spaces $\mathcal{N}(\tilde{T} - \lambda I)$ and $\mathcal{N}(\tilde{T}^H - \bar{\lambda} I)$ for these roots λ are given by (3.8a) and (3.8b), respectively. If the systems $(T_1 - \lambda I)^{-1} e_k$ and $(T_2 - \lambda I)^{-1} e_1$ can be solved stably by Gaussian elimination with partial pivoting, $Z(\lambda)$ in (6.4) can be computed in $O(n)$ operations for each λ and these systems can be solved simultaneously. Likewise, $Z'(\lambda) = -v^T (T - \lambda)^{-2} u$ is required for a Newton-type rootfinder. This can also be evaluated in $O(n)$ operations for each λ.

We now look in more detail how to accomplish the three tasks in the preceding paragraph. Let m_j denote the geometric multiplicity of λ_j as an eigenvalue of T, and let d denote the number of distinct eigenvalues of T. Then, when T is an $n \times n$ matrix, we need to do the following:

1. Find $\sigma(T)$, $W_R^j = \mathcal{N}(T - \lambda_j I)$, $W_L^j = \mathcal{N}(T - \lambda_j I)^H$, $j = 1, .., d$.

   $\sigma(T)$: Since T is block diagonal, $\sigma(T) = \sigma(T_1) \cup \sigma(T_2)$. If T_1 and T_2 are the leaves of the recursion tree, we can compute their eigenvalues by two calls to some routine, EVALUE, that computes the eigenvalues of a non-symmetric tridiagonal matrix. If we continue to recurse the algorithm, these eigenvalues are found instead from the subproblems of T_1 and T_2.

   W_R^j, W_L^j : If T_1 and T_2 are at the leaves of the recursion tree, one way to find these null spaces is to use the SVD. Let $(T_1 - \lambda_j I) = L_1 \Sigma_1 R_1^H$ and $(T_2 - \lambda_j I) = L_2 \Sigma_2 R_2^H$ be SVD's. Then

   $$T - \lambda_j I = \begin{pmatrix} L_1 & 0 \\ 0 & L_2 \end{pmatrix} P P^T \begin{pmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{pmatrix} P P^T \begin{pmatrix} R_1^H & 0 \\ 0 & R_2^H \end{pmatrix}$$

   where P is a permutation matrix that orders the singular values of Σ_1 and Σ_2 putting all the zeroes in the last m_j diagonal positions of $\Sigma = P^T \begin{pmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{pmatrix} P$. If $L = \begin{pmatrix} L_1 & 0 \\ 0 & L_2 \end{pmatrix} P$ and $R^H = P^T \begin{pmatrix} R_1^H & 0 \\ 0 & R_2^H \end{pmatrix}$, we see that $L \Sigma R^H$ is a SVD of $T - \lambda_j I$ and that W_R^j has as columns the last m_j columns of R. Likewise W_L^j has as columns the last m_j columns of L. We note that if m_{j1}, the multiplicity of λ_j as eigenvalue of T_1 is zero, then we do not have to perform the SVD associated with $T_1 - \lambda_j I$. Likewise if m_{j2}, the multiplicity of λ_j as eigenvalue of T_2 is zero, then we do not have to perform the SVD associated with $T_2 - \lambda_j I$. We also note that the entire SVD is not needed since we only require the parts associated with the zero singular values whose geometric multiplicity we know. An alternative approach calculating a partial SVD is to use inverse subspace iteration. If T_1 and T_2 are not at the leaves of the recursion tree, these null spaces are determined instead by the theory using the associated subproblems.

2. Find $\mathcal{N}(Z_e(\lambda_j))$, $\mathcal{N}(\tilde{T} - \lambda_j)$, and $\mathcal{N}(\tilde{T} - \lambda_j)^H$, $j = 1, ..., d$.

   $\mathcal{N}(Z_e(\lambda_j))$: First we must calculate $Z_e(\lambda_j)$. To do this, we need to calculate $W_L^{jH} u$ and $v^T W_R^j$ which can be done from the output of step 1. Next, we calculate the $m - 1$ columns of $W_R Q_R^H$ and $W_L Q_L^H$ which are the deflated right and left eigenvectors, respectively. Finally, if $\sigma\pi = 0$ we must calculate $v^T(\lambda_j I - T)^+ u$. Inspired by the fact that $(I - W_R^j W_R^{jH})(\mu_j - T)^{-1}(I - W_L^j W_L^{jH})$ tends *continuously* to $(\lambda_j - T)^+$ as $\mu \to \lambda_j$ [6, p.218], we first project u on the range of $(\lambda_j - T)$ by $(I - W_L^j W_L^{jH})$. Then we solve $(\lambda_j - T)t = (I - W_L^j W_L^{jH})u$ formally by Gaussian elimination replacing zero pivots by tiny numbers. Finally we project t on the orthogonal complement of the nullspace of $T - \lambda_j$. Since we know λ_j, W_L^j, and W_R^j this can be done in $O(m_j n)$ work.

$\mathcal{N}(\tilde{T}-\lambda_j), \mathcal{N}(\tilde{T}-\lambda_j)^H$: We use the five cases from section 4.2.

3. Find $\sigma(\tilde{T})\backslash\sigma(T)$. We must find the roots of the secular equation (6.4) using a Newton-type rootfinder. Then the associated left and right eigenvectors associated with these roots are calculated by (3.8a) and (3.8b).

The process described above can be recursed. We now provide a recursive algorithm for the nonsymmetric tridiagonal problem and give the associated costs. From the costs, it can be seen which difficulties must be overcome in order to have a functional algorithm.

Given an $n \times n$ matrix $\tilde{T} = T + uv^T$, where $T = T_1 \oplus T_2$, the following recursive subroutine DC finds $(\tilde{\lambda}_j, \tilde{m}_j, (\tilde{W}_L^j, \tilde{W}_R^j) \in \mathbb{C}^{n\times\tilde{m}_j}, j = 1, 2, ..., \tilde{d})$, the $\tilde{d}$ distinct eigenvalues $\tilde{\lambda}_j$ together with their associated left and right eigenspaces. SPLIT calculates u and v to split the problem $\tilde{T}$, and COMBINE organizes the results of the two previous subproblems, T_1 and T_2, to find the eigenvalues, multiplicities, and eigenspaces of T. FORMZ uses (3.16) to calculate the entries in $Z_e(\lambda_j) \in \mathbb{C}^{(1+m_j)\times(1+m_j)}$. FORM$\hat{Z}$ uses (4.1) to calculate the entries in $\hat{Z}_e(\lambda_j) \in \mathbb{C}^{(1+m_j)\times(1+m_j)}$ and hence $\hat{Z}_{e,11}(\lambda_j)$ in (4.6) or (4.7). LEFT and RIGHT calculate the respective eigenspaces using the five cases given in section 4.2, resulting in $\tilde{W}_L^j, \tilde{W}_R^j \in \mathbb{C}^{n\times\tilde{m}_j}$. ROOTS finds the s distinct roots of (6.4) and RIGHTR and LEFTR use (3.8a) and (3.8b), respectively, to find the eigenspaces associated with these roots, resulting in $\tilde{W}_R^j, \tilde{W}_L^j \in \mathbb{C}^{n\times 1}$.

SUB DC($\tilde{T}, n, (\tilde{\lambda}_j, \tilde{m}_j, \tilde{W}_L^j, \tilde{W}_R^j, j = 1, 2, ..., \tilde{d})$)

If $\tilde{T}$ is still too big
then SPLIT$(T_1, T_2, u, v, n_1, n_2, n)$
DC$(T_1, n_1, (\lambda_{j1}, m_{j1}, W_{L1}^j, W_{R1}^j, j = 1, 2, ..., d_1))$
DC$(T_2, n_2, (\lambda_{j2}, m_{j2}, W_{L2}^j, W_{R2}^j, j = 1, 2, ..., d_2))$
COMBINE$(\lambda_j, m_j, W_L^j, W_R^j, j = 1, ..., d)$
(* We now have the d distinct eigenvalues of
T and their respective geometric multiplicities
and left and right eigenspaces. *)

$\tilde{d} = 0$
For $j = 1, d$ do
FORMZ $(Z_e(\lambda_j))$
FORM$\hat{Z}(\hat{Z}_e(\lambda_j))$
If $(m_j > 1$ or $\pi\phi = 0)$ **then**
LEFT$(\tilde{W}_L^j)$
RIGHT$(\tilde{W}_R^j)$
$\tilde{d} = \tilde{d} + 1$
End(If)
End(For)
(* We now have the eigenvalues, associated geometric
multiplicities, and left and right eigenspaces of $\tilde{T}$
corresponding to the case that these eigenvalues are
also eigenvalues of T. *)

ROOTS($\tilde{\lambda}_j, j = \tilde{d}+1, ..., \tilde{d}+s$)
For $j = \tilde{d}+1, \tilde{d}+s$ do
LEFTR($\tilde{W}_L^j$)
RIGHTR($\tilde{W}_R^j$)
End(For)
$\tilde{d} = \tilde{d} + s$
(* We now have all the $\tilde{d}$ distinct eigenvalues of $\tilde{T}$ and their associated geometric multiplicities, $\tilde{m}_j$, and left and right eigenspaces. *)

else EVALUE($\tilde{T}$) (* Gives $\tilde{\lambda}_j, j = 1, \tilde{d}$ *)
SVD($\tilde{T} - \tilde{\lambda}_j I, \tilde{m}_j, j = 1, \tilde{d}$)
(*We now have all the $\tilde{d}$ distinct evalues of $\tilde{T}$, the associated geometric multiplicities, $\tilde{m}_j$, and the left and right eigenspaces. We get to this **else** clause when $\tilde{T}$ is a leaf of the recursion tree. This is a partial SVD. *)
End(If)

End(SUB DC)

We now consider the cost of algorithm DC on both sequential and parallel machines. For simplicity, we will not count the communication cost in the parallel version, but instead report the best possible that one could do with a reasonable level of granularity.

We denote by N_k the quantity $\frac{N}{2^k}$. For i levels of recursion, we execute the main **else** clause 2^i times for the $\tilde{T}$ problems at the leaves of the recursion tree. For the l-th problem on the i-th recursion level, we must do partial SVD's (or some other black box that can find null spaces) on matrices of size $N_i \times N_i$ in order to find the $\tilde{d}_{il}$ left and right null spaces. The j-th SVD for the l-th problem on level i finds the null spaces associated with λ_j of dimension $m_{j,il}$. Since these $\tilde{T}$'s are unreduced tridiagonal matrices $m_{j,kl} = 1$. The cost to do one partial SVD includes a reduction to bidiagonal form which takes $O(N_i^2)$ and the calculation of the partial SVD of this form which takes $O(N_i^2)$ as well. The cost of one EVALUE problem also requires the same order of operations. We assume all leaves of the tree can execute in parallel as well as all SVD's at each leaf. The total costs are given below:

$$\text{Sequential: } 2^i \text{ COST(EVALUE, } N_i) + \sum_{l=1}^{2^i} \sum_{j=1}^{\tilde{d}_{il}} \text{COST(SVD,}N_i,m_{j,il}) \leq c\frac{N^3}{2^{2i}}$$

$$\text{Parallel: } \quad \text{COST(EVALUE, } N_i) + \text{COST(SVD, } N_i, \max_{j,l} m_{j,il}) \leq c\frac{N^2}{2^{2i}}$$

If $i = \log_2 N - 1$, we find the eigenpairs of 2×2 matrices and the above costs become $25\log_2 N$ and 25, respectively.

On recursion level k, the main **then** clause is executed 2^k times for matrices $\tilde{T}$ of size N_k (k ranges from $i-1$ down to 0 with 0 being the root of the tree).

For each problem on level k, we must combine the results of its two subproblems to get the associated eigenvalues and eigenspaces of $T \in \mathbb{C}^{N_k \times N_k}$ where $T = T_1 \oplus T_2$. The routine COMBINE executes for free, since $\left(\lambda_{j1}, \begin{pmatrix} W_{R1}^j \\ 0 \end{pmatrix}\right), j = 1, ..., d_1$, and $\left(\lambda_{j2}, \begin{pmatrix} 0 \\ W_{R2}^j \end{pmatrix}\right), j = 1, ..., d_2$, are right eigenpairs of T. Likewise, a similar statement holds for the left eigenpairs of T.

For the l-th problem on level k, we do d_{lk} FORMZ's on matrices of size N_k. The j-th FORMZ is for an eigenvalue with geometric multiplicity $m_{j,kl}$. Since for each problem T is block diagonal consisting of two irreducible tridiagonal matrices as diagonal blocks, we know $m_{j,kl} \leq 2$. COST(FORMZ, N_k, $m_{j,kl}$) can be split into two parts. The first part is the cost to form $W_L^H u$ and $v^T W_R$ where u and v are seen from (6.3) to be special N_k vectors with at most two nonzero elements and W_L and W_R are $N_k \times m_{j,kl}$ matrices. Hence, both these multiplications together only require 12 flops. The second part of the cost is to form $1 - v^T(\lambda - T)^+ u$ where T is $N_k \times N_k$ consisting of two diagonal blocks. As described earlier, this can be done with $cm_{j,kl}N_k$ work. The total costs are given below.

Sequential:

$$\sum_{k=0}^{i-1} \sum_{l=1}^{2^k} \sum_{j=1}^{d_{kl}} \text{COST(FORMZ, } N_k, m_{j,kl}) \leq 12iN + \sum_{k=0}^{i-1} \sum_{l=1}^{2^k} \sum_{j=1}^{d_{kl}} \text{COST}(1 - v^T(\lambda_j - T)^+ u, N_k, m_{j,kl}) \leq cN^2$$

Parallel:

$$\sum_{k=0}^{i-1} \text{COST(FORMZ, } N_k, \max_{j,l} m_{j,kl}) \leq 12i + \sum_{k=0}^{i-1} \text{COST}(1 - v^T(\lambda - T)^+ u, N_k, \max_{j,l} m_{j,kl}) \leq cN$$

We note that the calculation of $1 - v^T(\lambda - T)^+ u$ in the above expression only needs to be done for problems which satisfy $\pi\phi = 0$. Hence, the costs above are most likely $12iN$ and $12i$, respectively.

For every FORMZ above, we do a FORM$\hat{Z}$. For each FORM$\hat{Z}$ two reflectors must be calculated. Hence COST(FORM$\hat{Z}$, N_k, $m_{j,kl}$) is only $cm_{j,kl}$, with $\max_{j,l} m_{j,kl} \leq 2$. The total costs are given below.

Sequential: $\sum_{k=0}^{i-1} \sum_{l=1}^{2^k} \sum_{j=1}^{d_{kl}} \text{COST(FORM}\hat{Z}, N_k, m_{j,kl}) \leq c(iN)$

Parallel: $\sum_{k=0}^{i-1} \text{COST(FORM}\hat{Z}, N_k, \max_{j,l} m_{j,kl}) \leq ci$

Now, let $\bar{d}_{kl}$ denote the number of distinct eigenvalues of T of problem l on recursion level k that are actually eigenvalues of the associated matrix $\tilde{T}$. We need to

find COST(LEFT, N_k) and COST(RIGHT, N_k). From the 5 cases in section 4.2, we must form the m columns of $W_R Q_R^H$ and $W_L Q_L^H$ where $W_R, W_L \in \mathbb{C}^{N_k \times m_{j,kl}}$ and $Q_R, Q_L \in \mathbb{C}^{m_{j,kl} \times m_{j,kl}}$. Together, these calculations cost $O(N_k m_{j,kl}^2)$ work. At this point, we note that $Z_e^H(\lambda_j)$ and $\hat{Z}_e^H(\lambda_j)$ are needed in order to find LEFT($\tilde{W}_L^j$). These are simply gotten from $Z_e(\lambda_j)$ and $\hat{Z}_e(\lambda_j)$ free of cost. We observe that the values $\tilde{W}_R^j$ from RIGHT involving the pseudoinverse are gotten from the five cases in section 4.2 for free once FORM$\hat{Z}$ is done since $(T-\lambda_j)^+ u$ must be calculated anyway in order to do FORM$\hat{Z}$. However, $(T^H - \bar{\lambda}_j)^+ v$ need not be known in order to calculate $\hat{Z}_e^H(\lambda_j)$. Hence, from the five cases, we see that COST(LEFT, N_k) also involves the calculation of at most one $(T^H - \bar{\lambda})^+ v$ for a possible m-th or $m+1$-st left eigenvector. Since $T \in \mathbb{C}^{N_k \times N_k}$ this takes $c m_{j,kl} N_k$ work where $m_{j,kl} \leq 2$. The total costs are given below.

Sequential:

$$\sum_{k=0}^{i-1} \sum_{l=1}^{2^k} \sum_{j=1}^{\tilde{d}_{kl}} (\text{COST(LEFT}, N_k) + \text{COST(RIGHT}, N_k))$$
$$\leq \sum_{k=0}^{i-1} \sum_{l=1}^{2^k} \sum_{j=1}^{\tilde{d}_{kl}} \text{COST}((T^H - \bar{\lambda}_j)^+ v, N_k, m_{j,kl}) + \sum_{k=0}^{i-1} \sum_{l=1}^{2^k} \sum_{j=1}^{\tilde{d}_{kl}} c(N_k m_{j,kl}^2)$$
$$\leq cN^2$$

Parallel:

$$\sum_{k=0}^{i-1} (\text{COST(LEFT}, N_k) + \text{COST(RIGHT}, N_k))$$
$$\leq \sum_{k=0}^{i-1} \text{COST}((T^H - \bar{\lambda}_j)^+ v, N_k, \max_{j,l} m_{j,kl}) + \sum_{k=0}^{i-1} c(N_k m_{j,kl}^2) \leq cN$$

From the five cases in section 4.2, we see that the m-th or $m+1$-st left eigenvector only occurs when $\pi\phi = 0$. Hence, most likely the costs for the pseudoinverse calculation above are negligible but the total costs still have the same complexity.

We now have $s_{kl} = \tilde{d}_{kl} - \bar{d}_{kl}$ remaining distinct eigenvalues of the l-th $\tilde{T}$ problem on recursion level k. Using a Newton-type rootfinder and assuming the number of steps per root is bounded by a constant, the cost to find the s_{kl} roots of the l-th problem on level k is $cs_{kl}N_k$ on a sequential machine. For the associated parallel cost, we note that all 2^k rootfinding problems on level k can be done simultaneously, and that each of these problems in turn can utilize $p_{kl} \leq s_{kl}$ processors. The total costs are given below for the case $p_{kl} = s_{kl}$.

Sequential: $\sum_{k=0}^{i-1} \sum_{l=1}^{2^k} \text{COST(ROOTS, } s_{kl}, N_k) \leq cN^2$

Parallel: $\sum_{k=0}^{i-1} \text{COST(PROOTS,}s_{kl}, N_k, p_{kl}) \leq cN$

We now find the associated right and left eigenspaces corresponding to these roots. COST(RIGHTR,N_k) becomes COST($(T-\lambda)^{-1}u$, N_k) and COST(LEFTR,N_k) becomes COST($(T^H - \bar{\lambda})^{-1} v$,$N_k$). Since the matrices T and T^H are tridiagonal, each of these calculations can be done in $17N_k$ flops (adds and multiplies) using Gaussian

elimination with partial pivoting [17, p.41] assuming they can be done stably. If so, the total costs are given below.

$$\text{Sequential: } \sum_{k=0}^{i-1}\sum_{l=1}^{2^k} s_{kl}\,(\text{COST(LEFTR, } N_k)+\text{COST(RIGHTR, } N_k)) \approx 64\sum_{k=0}^{i-1} N_k \sum_{l=1}^{2^k} s_{kl} \le 64N^2$$

$$\text{Parallel: } \sum_{k=0}^{i-1} (\ \text{COST(LEFTR, } N_k) + \text{COST(RIGHTR, } N_k)\ ) \approx 64\sum_{k=0}^{i-1} N_k \le 64N$$

The above upper bounds of $O(N^2)$ in the sequential case and $O(N)$ in the parallel case may seem surprising, since it is usually reported in the literature that the eigenspaces are gotten in $O(N^3)$ sequential work and $O(N^2)$ parallel work. For the symmetric tridiagonal problem, these calculations are usually done by first forming the decomposition $(T-\lambda)^{-1} = Q(\Lambda-\lambda)^{-1}Q^T$ and then multiplying these three matrices by u. This procedure is perhaps more stable, but indeed results in the higher complexity because of the sparsity structure of Q. In our nonsymmetric case, if we had the decomposition $(T-\lambda)^{-1} = X\Delta X^{-1}$ including X^{-1}, where Δ is block diagonal, then we would also report $O(N^3)$ sequential and $O(N^2)$ parallel complexity if the tridiagonal solve directly with the matrix $(T-\lambda)^{-1}$ could not be done in a stable manner. We note that if we calculate the principal vectors as well as the eigenspaces, we would have a decomposition of this form (cf. section 3).

From the above costs, we see that the sequential algorithm requires $O(\frac{N^3}{2^{2i}} + N^2)$ time, assuming all the calculations can be done stably. Likewise, our calculations show the parallel algorithm requires $O(\frac{N^2}{2^{2i}} + N)$ time. So, for a parallel algorithm of $O(N)$ to result, we must recurse the problem until $i = O(\frac{1}{2}\log_2 N)$.

It is also interesting to point out that these results are obtained *without* creating a smaller problem as a result of deflation. Hence, if stability is not an issue, there is no motivation to create a smaller problem since working with the original $N \times N$ matrix T and solving systems with $T-\lambda$ is good enough.

6.2. The Hessenberg eigenvalue problem. In this section we derive formulae upon which a divide and conquer algorithm for the eigenvalue problem

$$\tilde{H}x = \lambda x, \tag{6.5}$$

could be based. In (6.5) $\tilde{H}$ is an unreduced upper Hessenberg matrix. We define the matrix H to be

$$H := \tilde{H} - h_{k+1,k}\begin{pmatrix} p \\ e_1 \end{pmatrix}\begin{pmatrix} e_k \\ q \end{pmatrix}^T = \tilde{H} + uv^T, \qquad p \in \mathbb{R}^k, q \in \mathbb{R}^{n-k}, \tag{6.6}$$

where p and q are *arbitrary* vectors. We thus can consider $\tilde{H}$ to be a rank-1-modification of H! H has a zero entry at position $(k+1,k)$.

Let

$$H = \begin{pmatrix} H_1 & H_{12} \\ 0 & H_2 \end{pmatrix}, \tag{6.7}$$

where $H_1 \in \mathbb{R}^{k\times k}$ and $H_2 \in \mathbb{R}^{k'\times k'}$ are unreduced Hessenberg matrices and $H_{12} \in \mathbb{R}^{k\times k'}$.

A divide and conquer algorithm would proceed in such a way that first the eigenvalues, λ, of H and associated spaces $W_R = \mathcal{N}(H-\lambda I)$ and $W_L = \mathcal{N}(H^H - \bar{\lambda} I)$ are computed. Then the extended matrix $Z_e(\lambda)$ in (3.16) with $\tilde{A} = \tilde{H}$, $A = H$, $V^T = (e_k^T \;\; q^T)$, and $U^T = h_{k+1,k}(p^T \;\; e_1^T)$ must be calculated. If a nontrivial null space for $Z_e(\lambda)$ exists, then we find $\mathcal{N}(\tilde{H}-\lambda I)$ and $\mathcal{N}(\tilde{H}^H - \bar{\lambda} I)$ from the five cases in section 4.2. Next, a Newton-type root finding scheme can be used to find eigenvalues, λ, of $\tilde{H}$ that are not eigenvalues of H. To this end, the determinant of $Z(\lambda)$ of (3.3) has to be investigated and the λ's that make this determinant zero must be found. For Hessenberg matrices, $Z(\lambda), \lambda \notin \sigma(H)$, is the scalar,

$$
\begin{aligned}
Z(\lambda) &= 1 + \alpha \begin{pmatrix} e_k \\ q \end{pmatrix}^T (H-\lambda)^{-1} \begin{pmatrix} p \\ e_1 \end{pmatrix}, \qquad \alpha = h_{k+1,k}, \\
&= 1 + \alpha \left[e_k^T (H_1-\lambda)^{-1} p - e_k^T (H_1-\lambda)^{-1} H_{12} (H_2-\lambda)^{-1} e_1 \right. \\
&\qquad \left. + q^T (H_2-\lambda)^{-1} e_1 \right]
\end{aligned} \tag{6.8}
$$

The associated spaces $\mathcal{N}(\tilde{H}-\lambda I)$ and $\mathcal{N}(\tilde{H}^H - \bar{\lambda} I)$ for these roots λ are given by (3.8a) and (3.8b).

Note that the complexity to form $Z(\lambda)$ does not increase essentially if p and q are nonzero, because the vectors $(H_1^T - \lambda)^{-1} e_k$ and $(H_2 - \lambda)^{-1} e_1$ have to be computed anyway! Assuming these systems can be stably solved, the computation of $Z(\lambda)$ costs $\frac{n^2}{2} + O(n)$ flops for each λ if one assumes that $k \approx k' \approx \frac{n}{2}$. On a parallel machine, the solution of these Hessenberg systems can be done simultaneously. Likewise, the evaluation of $Z'(\lambda) = -v^T (H-\lambda)^{-2} u$ can be done in $n^2 + O(n)$ operations.

We now look in more detail how to accomplish the three tasks in the preceding paragraph. Let m_j denote the geometric multiplicity of λ_j as an eigenvalue of H, and let d denote the number of distinct eigenvalues of H. Then, when H is an $n \times n$ matrix, we need to do the following:

1. Find $\sigma(H)$, $W_R^j = \mathcal{N}(H-\lambda_j I)$, $W_L^j = \mathcal{N}(H-\lambda_j I)^H$, $j = 1, .., d$.

   $\sigma(H)$: Since $H = \begin{pmatrix} H_1 & H_{12} \\ 0 & H_2 \end{pmatrix}$, $\sigma(H) = \sigma(H_1) \cup \sigma(H_2)$. If H_1 and H_2 are the leaves of the recursion tree, we can compute these eigenvalues by two calls to some routine, EVALUE, that computes the eigenvalues of a Hessenberg matrix.

   W_R^j, W_L^j : The first step toward finding these eigenspaces of H is to find the eigenspaces of H_1 and H_2. If H_1 and H_2 are at the leaves of the recursion tree, one way to find these null spaces is to find partial SVDs of $(H_1 - \lambda_j I)$ and $(H_2 - \lambda_j I)$. To complete the process, we must devise the strategy used in the COMBINE routine to find the eigenspaces of H from those of H_1 and H_2. Recall in the tridiagonal case, this was trivial and required no cost. We proceed as follows. Since we want $\begin{pmatrix} H_1 & H_{12} \\ 0 & H_2 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \lambda \begin{pmatrix} y_1 \\ y_2 \end{pmatrix}$, we have the equations

   $$H_1 y_1 + H_{12} y_2 = \lambda y_1 \tag{6.9a}$$

(6.9b) $$H_2 y_2 = \lambda y_2$$

Either $y_2 = 0$ or $y_2 \neq 0$. First suppose $y_2 = 0$. Then from (6.9a) we see that $H_1 y_1 = \lambda y_1$. That is, (λ, y_1) is an eigenvalue-right eigenvector pair of H_1. So $\left(\lambda, \begin{pmatrix} y_1 \\ 0 \end{pmatrix}\right)$ is an eigenvalue-right eigenvector pair of H. Now suppose $y_2 \neq 0$. Then from (6.9b) we see that $H_2 y_2 = \lambda y_2$. That is, (λ, y_2) must be an eigenvalue-right eigenvector pair of H_2. But (6.9a) must also be satisfied. If $H_{12} y_2 \in \mathcal{R}(H_1 - \lambda)$ then (6.9a) has one or more solutions. If λ is not an eigenvalue of H_1,

$$y_1 = -(H_1 - \lambda)^{-1} H_{12} y_2$$

is the unique solution, and $\left(\lambda, \begin{pmatrix} (\lambda - H_1)^{-1} H_{12} y_2 \\ y_2 \end{pmatrix}\right)$ is an eigenvalue-right eigenvector pair of H. If λ is also an eigenvalue of H_1 then we can check whether or not we have a solution by seeing if $W_{L1}^H H_{12} y_2 = 0$ where W_{L1} spans the left null space of $H_1 - \lambda$. If there are solutions, we can take the particular solution with smallest 2-norm

$$y_1 = (\lambda - H_1)^+ H_{12} y_2 \,.$$

Then $\left(\lambda, \begin{pmatrix} (\lambda - H_1)^+ H_{12} y_2 \\ y_2 \end{pmatrix}\right)$ is an eigenvalue-right eigenvector pair of H. The procedure for finding the eigenvalue-left eigenvector pairs parallels exactly this same procedure.

2. Find $\mathcal{N}(Z_e(\lambda_j))$, $\mathcal{N}(\tilde{H} - \lambda_j)$, and $\mathcal{N}(\tilde{H} - \lambda_j)^H$, $j = 1, ..., d$.

   $\mathcal{N}(Z_e(\lambda_j))$: First we must calculate $Z_e(\lambda_j)$. To do this, we need to calculate $W_L^H u$ and $v^T W_R$ which can be done from the output of step 1. Next, we must calculate the $m - 1$ columns of $W_R Q_R^H$ and $W_L Q_L^H$ which are the deflated right and left eigenvectors, respectively. Finally, if $\pi\phi = 0$, we must calculate $v^T(\lambda_j I - H)^+ u$. Inspired by the formula $\lim_{\mu_j \to \lambda_j}(I - W_R^j W_R^{j\,H})(\mu_j - H)^{-1}(I - W_L^j W_L^{j\,H})u = (\lambda_j - H)^+ u$, we first project u on the range of $(\lambda_j - H)$ by $(I - W_L^j W_L^{j\,H})$. Then we solve $(\lambda_j - H)h = (I - W_L^j W_L^{j\,H})u$ formally by Gaussian elimination replacing zero pivots by tiny numbers. Finally we project h on the orthogonal complement of the nullspace of $H - \lambda_j$. Since we know λ_j, W_R^j, and W_L^j we can do this in $O(n^2)$ time, where the dominant time is due to the solution of a Hessenberg system.

   $\mathcal{N}(\tilde{H} - \lambda_j)$, $\mathcal{N}(\tilde{H} - \lambda_j)^H$: Use the five cases from section 4.2.

3. Find $\sigma(\tilde{H}) \backslash \sigma(\tilde{H})$. We can find the roots of the secular equation (6.8) using a Newton-type rootfinder. Then the associated left and right eigenvectors associated with these roots are calculated by (3.8a) and (3.8b).

The process described above can be recursed. In fact, subroutine DC given in section 6.1 works for Hessenberg matrices with slight modifications. We replace $\tilde{T}$, T,

T_1, and T_2 by $\tilde{H}$, H, H_1, and H_2, respectively. The SPLIT routine also produces H_{12} which in turn must be taken into account in the COMBINE routine as described in the previous paragraphs. ROOTS and PROOTS use (6.8), the secular equation for the Hessenberg problem.

Following the previous discussion for the tridiagonal problem, we now give the cost of subroutine DC on both sequential and parallel machines.

The cost of one EVALUE problem for matrices of size N_i is $O(N_i^3)$ and one partial SVD also costs $O(N_i^3)$. The total costs are given below.

Sequential: 2^i COST(EVALUE, N_i) + $\sum_{l=1}^{2^i} \sum_{j=1}^{\tilde{d}_{il}}$ COST(SVD, N_i, $m_{j,il}$) $\leq c\frac{N^4}{2^{3i}}$

Parallel: COST(EVALUE, N_i) + COST(SVD, N_i, $\max_{j,l} m_{j,il}$) $\leq c\frac{N^3}{2^{3i}}$

In the sequential algorithm above, the dominant cost is due to the reduction of the Hessenberg matrices to bidiagonal form. As in the tridiagonal case, the cost of performing a partial SVD on a bidiagonal matrix of size $O(N_i)$ is $O(\frac{N^2}{2^{2i}})$.

For the l-th problem on level k, we must combine the results of its two subproblems to get the associated eigenvalues and eigenspaces of $H \in \mathbb{C}^{N_k \times N_k}$ where $H = \begin{pmatrix} H_1 & H_{12} \\ 0 & H_2 \end{pmatrix}$. From the previous discussion, COST(COMBINE, N_k) is $c_1\frac{N^2}{2^{3k}} + c_2$COST($(H_1 - \lambda)^+ y^*$) where y^* represents a $\frac{N}{k+1}$ vector. The total costs for the COMBINE routine are given below.

Sequential:

$$\sum_{k=0}^{i-1} \sum_{l=1}^{2^k} \text{COST(COMBINE}, N_k) \leq cN^2 + c_2 \sum_{k=0}^{i-1} 2^k \text{COST}((H_1 - \lambda)^+ y^*, \tfrac{N}{2^{k+1}}) \leq cN^2$$

Parallel:

$$\sum_{k=0}^{i-1} \text{COST(COMBINE}, N_k) \leq cN^2 + c_2 \sum_{k=0}^{i-1} \text{COST}((H_1 - \lambda)^+ y^*, \tfrac{N}{2^{k+1}}) \leq cN^2$$

COST(FORMZ, N_k, $m_{j,kl}$) can be split into two parts. The first part is identical to the tridiagonal case. The second part is the cost to form $1 - v^T(\lambda - H)^+ u$ where H is a $N_k \times N_k$ Hessenberg matrix. The cost of one EVALUE problem is $O(N_i^3)$ and one partial SVD also costs $O(N_i^3)$. The total costs are given below.

Sequential:

$$\sum_{k=0}^{i-1} \sum_{l=1}^{2^k} \sum_{j=1}^{d_{kl}} \text{COST(FORMZ}, N_k, m_{j,kl}) \leq 12iN + \sum_{k=0}^{i-1} \sum_{l=1}^{2^k} \sum_{j=1}^{d_{kl}} \text{COST}(1 - v^T(\lambda_j - H)^+ u, N_k) \leq cN^3$$

Parallel:

$$\sum_{k=0}^{i-1} \mathrm{COST}(\mathrm{FORMZ},\, N_k,\, \max_{j,l} m_{j,kl})$$
$$\leq 12i + \sum_{k=0}^{i-1} \mathrm{COST}(1 - v^T(\lambda - H)^+ u,\, N_k,\, \max_{j,l} m_{j,kl}) \leq cN^2$$

As in the tridiagonal case, the calculation of $1 - v^T(\lambda - H)^+ u$ is only done when $\pi\phi = 0$. Hence, most likely the costs above reduce to $12iN$ and $12i$, respectively.

The costs of FORM$\hat{Z}$ are identical to the tridiagonal case and are repeated below.

Sequential: $\sum_{k=0}^{i-1} \sum_{l=1}^{2^k} \sum_{j=1}^{d_{kl}} \mathrm{COST}(\mathrm{FORM}\hat{Z},\, N_k,\, m_{j,kl}) \leq c(iN)$

Parallel: $\sum_{k=0}^{i-1} \mathrm{COST}(\mathrm{FORM}\hat{Z},\, N_k,\, \max_{j,l} m_{j,kl}) \leq ci$

The costs of LEFT and RIGHT involving the m columns of $W_R Q_R^H$ and $W_L Q_L^H$ are the same as for the tridiagonal case. COST(RIGHT, N_k) involving the pseudoinverse is gotten for free during the formation of FORMZ and FORM$\hat{Z}$. From the five cases in section 4.2, we see that COST(LEFT, N_k) involving the pseduo inverse reduces to the calculation of at most one $(H^H - \bar{\lambda})^+ v$ for a possible m-th or $m+1$-st left eigenvector. This costs $O(\frac{N^2}{2^{2k}})$ work since H is a $N_k \times N_k$ Hessenberg matrix. The total costs are given below.

Sequential:
$$\sum_{k=0}^{i-1} (\sum_{l=1}^{2^k} \bar{d}_{kl}(\mathrm{COST}(\mathrm{LEFT}, N_k) + \mathrm{COST}(\mathrm{RIGHT}, N_k))$$
$$\leq \sum_{k=0}^{i-1} \sum_{l=1}^{2^k} \sum_{j=1}^{\bar{d}_{kl}} \mathrm{COST}((H^H - \bar{\lambda})^+ v, N_k) + O(N^2) \leq cN^3$$

Parallel:
$$\sum_{k=0}^{i-1} (\mathrm{COST}(\mathrm{LEFT}, N_k) + \mathrm{COST}(\mathrm{RIGHT}, N_k))$$
$$\leq \sum_{k=0}^{i-1} \mathrm{COST}((H^H - \bar{\lambda})^+ v, N_k) + O(N) \leq cN^2$$

As in the tridiagonal case, the calculations above are only done for problems which satisfy $\pi\phi = 0$. So, most likely the costs involving the pseudoinverse reduce to zero and the total costs become $O(N^2)$ and $O(N)$, respectively.

We now have $s_{kl} = \tilde{d}_{kl} - \bar{d}_{kl}$ remaining distinct eigenvalues of the l-th $\tilde{H}$- problem on recursion level k. We use a Newton-type rootfinder and assume that a constant number of steps per root are required. The total costs are given below when $p_{kl} = s_{kl}$.

Sequential: $\sum_{k=0}^{i-1} \sum_{l=1}^{2^k} \mathrm{COST}(\mathrm{ROOTS},\, s_{kl},\, N_k\,) \leq cN^3$

Parallel: $\sum_{k=0}^{i-1} \text{COST(PROOTS,}s_{kl}, N_k, p_{kl}) \leq cN^2$

We now find the eigenspaces corresponding to these roots. COST(RIGHTR, N_k) and COST(LEFTR, N_k) become COST($(H-\lambda)^{-1}u$, N_k) and COST($(H^H-\bar{\lambda})^{-1}v$, N_k), respectively. Since the matrices H and H^H are Hessenberg, both calculations can be done in $O(\frac{N^2}{4^k})$ time, assuming they can be done stably by solving linear systems. If so, the total costs are given below.

Sequential:

$$\sum_{k=0}^{i-1}\sum_{l=1}^{2^k} s_{kl}\,(\text{COST(LEFTR, } N_k) + \text{COST(RIGHTR, } N_k)) \leq c\sum_{k=0}^{i-1}\frac{N^2}{4^k}\sum_{l=1}^{2^k} s_{kl} \leq cN^3$$

Parallel:

$$\sum_{k=0}^{i-1}(\text{COST(LEFTR, } N_k) + \text{COST(RIGHTR, } N_k)) \leq c\sum_{k=0}^{i-1}\frac{N^2}{4^k} \leq cN^2$$

From the above costs, we see that the sequential algorithm requires $O(\frac{N^4}{2^{3i}} + N^3)$ time, assuming all the calculations can be done stably. Likewise, our results show the parallel algorithm takes $O(\frac{N^3}{2^{3i}} + N^2)$ time. So for a parallel algorithm of $O(N^2)$ to result, we must recurse the problem until $O(i = \frac{1}{3}\log_2 N)$.

As in the tridiagonal case, parallel results of one order less in N can be obtained, assuming stability, *without* creating smaller problems after the deflation process. On sequential computers, however, one may improve the constant in the complexity result by creating smaller problems in the diagonalizable case. We feel a better path is to abandon the Hessenberg form if possible and reduce the original matrix to another sparser form, perhaps tridiagonal, in a stable way.

In this section, we have demonstrated progress toward the development of a feasible divide and conquer recursive algorithm for both nonsymmetric tridiagonal and Hessenberg matrices. Without making any assumptions on diagonalizability, our algorithm finds all eigenvalues and left and right eigenvectors in $O(N^3)$ and $O(N^2)$ sequential time and $O(N^2)$ and $O(N)$ parallel time for Hessenberg and tridiagonal matrices, respectively. In contrast, the QR algorithm requires $O(N^3)$ work on a sequential machine to find all eigenvalues and Schur vectors for both Hessenberg and tridiagonal matrices. Also, in the symmetric case, the usual implementation of Cuppen's method for tridiagonal matrices requires $O(N^3)$ work to find all eigenpairs, although we observe that the eigenvectors could be gotten by solving tridiagonal systems (as we do in the nonsymmetric case) with $O(N^2)$ work. This observation is based on the assumption that these calculations can be done stably.

7. Conclusions. In this paper we have derived the theory needed to devise divide and conquer algorithms for nonsymmetric eigenvalue problems. We have formulated algorithms for solving eigenvalue problems involving tridiagonal and Hessenberg matrices. Especially the tridiagonal divide and conquer algorithm has a very favorable complexity: $O(N^2)$ and $O(N)$ for the computation of all eigenvalues and vectors on sequential and parallel computers with $O(N)$ processors, respectively.

As had to be expected, the cost for the Hessenberg divide and conquer algorithm is one power of N higher. So, the divide and conquer algorithm has the same complexity as the Hessenberg QR algorithm on serial computers, but can be parallelized. But recall, the divide and conquer and QR algorithms do not compute the same decompositions. From the considerations at the end of section 3 it is clear that one should not compute the Schur decomposition by a divide and conquer algorithm.

The Hessenberg form is certainly not the matrix form we want to use in a divide and conquer algorithm. It is appropriate for the stable computation of the Schur decomposition by the QR algorithm since general matrices can be stably transformed into this form.

The difficulty of stably tridiagonalizing arbitrary matrices prevents a widespread use of nonsymmetric tridiagonal eigenproblem solvers. It may be possible to transform arbitrary matrices stably into a a very sparse but no longer tridiagonal form [18]. The divide and conquer algorithm could eventually take advantage of this form as systems of linear equations with such matrices have to be formed in the course of the algorithm.

Our approach is a theoretical one. We do not address important practical issues as

- *Numerical multiplicity of the eigenvalues.* In real arithmetic, $m_{j,kl}$ may be greater than 2. This could substantially degrade the costs of the routines SVD, FORMZ, LEFT, and RIGHT, especially on parallel machines since the time is governed by $\max_{j,l} m_{j,kl}$.
- *Zero finding.* There are two very difficult problems connected with zerofinding. We do not know *how many* zeros, i.e. eigenvalues, we have to look for nor do we know *where* they are. Jessup [15] discusses zerofinders which are based on the principle of the argument (Cauchy's integral formula). As does Jessup, we doubt their practicability. The calculation of the integrals involved is too expensive. One probably has to resort to Newton's iteration. As with any zerofinder, in order not to find the same zero several times, it may be necessary to successively deflate zeros which ruins parallelism.
- *Stability.* It is yet to be seen whether the calculation of the eigenvectors by solving linear systems of the form $(\lambda - T)^{-1}$ or $(\lambda - H)^{-1}$ will give numerically linearly independent eigenvectors for close values of λ. A related issue is the calculation of $(\lambda - T)^{+}u$. There may be cancellation when projecting onto the orthogonal complements of the null spaces. It is possible that a well conditioned problem $\tilde{T}$ or $\tilde{H}$ can have ill conditioned subproblems T_1, T_2 or H_1, H_2, respectively. This problem of nonsymmetric divide and conquer methods as well as other problems are discussed in [15] for a particular rank-2 splitting.

From this short list one sees that there are several problems to be solved before a practical implementation of a divide and conquer algorithm for the nonsymmetric problem becomes possible.

REFERENCES

[1] G. Ammar, L. Reichel, and D. Sorensen, *An implementation of a divide and conquer method for the unitary eigenproblem*, To appear in ACM Trans. Math. Softw.

[2] P. ARBENZ, W. GANDER, AND G. H. GOLUB, *Restricted rank modification of the symmetric eigenvalue problem: Theoretical considerations*, Linear Algebra Appl., 104 (1988), pp. 75–95.
[3] P. ARBENZ AND G. H. GOLUB, *On the spectral decomposition of Hermitian matrices modified by low rank perturbations with applications*, SIAM J. Matrix Anal. Appl., 9 (1988), pp. 40–58.
[4] C. BEATTIE AND D. FOX, *Schur complements and the Weinstein–Aronszajn theory for modified matrix eigenvalue problems*, Linear Algebra Appl., 108 (1988), pp. 37–61.
[5] J. R. BUNCH, C. P. NIELSON, AND D. C. SORENSEN, *Rank-one modification of the symmetric eigenproblem*, Numer. Math., 31 (1978), pp. 31–48.
[6] S. L. CAMPBELL AND C. D. MEYER, *Generalized Inverses of Linear Transformations*, Surveys and Reference Works in Mathematics 4, Pitman, London, 1979.
[7] J. J. M. CUPPEN, *A divide and conquer method for the symmetric tridiagonal eigenproblem*, Numer. Math., 36 (1981), pp. 177–195.
[8] J. J. DONGARRA AND D. C. SORENSEN, *A fully parallel algorithm for the symmetric eigenvalue problem*, SIAM J. Sci. Stat. Comput., 8 (1987), pp. s139–s154.
[9] R. FLETCHER AND D. C. SORENSEN, *An algorithmic derivation of the Jordan canonical form*, Amer. Math. Monthly, 90 (1983), pp. 12–16.
[10] K. GATES, *Using inverse iteration to improve the divide and conquer algorithm*, Tech. Report 159, ETH Zürich, Departement Informatik, May 1991.
[11] G. A. GEIST, *Reduction of a general matrix to tridiagonal form*, SIAM J. Matrix Anal. Appl., 12 (1991), pp. 362–373.
[12] G. H. GOLUB, *Some modified matrix eigenvalue problems*, SIAM Rev., 15 (1973), pp. 318–334.
[13] G. H. GOLUB AND C. F. VAN LOAN, *Matrix Computations*, The Johns Hopkins University Press, Baltimore, MD, 2nd ed., 1989.
[14] S. H. GOULD, *Variational Methods for Eigenvalue Problems*, University of Toronto Press, Toronto, 2nd ed., 1966.
[15] E. R. JESSUP, *A case against a divide and conquer approach to the nonsymmetric eigenvalue problem*, Tech. Report (in preparation), Department of Computer Science, University of Colorado, Boulder, CO.
[16] E. R. JESSUP AND D. C. SORENSEN, *A parallel algorithm for computing the singular value decomposition of a matrix*, Techn. Memo. 102, Argonne National Laboratory, Mathematics and Computer Science Division, December 1987.
[17] H. R. SCHWARZ, *Numerical Analysis*, John Wiley, Chichester, 1989.
[18] W.-P. TANG, *A stabilized algorithm for tridiagonalization of an unsymmetric matrix*, Tech. Report CS-88-14, University of Waterloo, Department of Computer Science, April 1988.
[19] A. WEINSTEIN AND W. STENGER, *Methods of Intermediate Problems for Eigenvalues*, Academic Press, New York, 1972.
[20] J. H. WILKINSON, *The Algebraic Eigenvalue Problem*, Clarendon Press, Oxford, 1965.

Gelbe Berichte des Departements Informatik

147	G. Weikum, P. Zabback, P. Scheuermann	Dynamic File Allocation in Disk Arrays (vergriffen)
148	D. Degiorgi	A New Linear Algorithm to Detect a Line Graph and Output its Root Graph
149	A. Moenkeberg, G. Weikum	Conflict-Driven Load Control for the Avoidance of Data-Contention Thrashing
150	M.H. Scholl, Ch. Laasch M. Tresch	Updatable Views in Object-Oriented Databases (vergriffen)
151	C. Szyperski	Write - An extensible Text Editor for the Oberon System
152	M. Bronstein	On Solutions of Linear Ordinary Difierential Equations in their Coefficient Field
153	G.H. Gonnet, D.W. Gruntz	Algebraic Manipulation: Systems (vergriffen)
154	G.H. Gonnet, St.A. Benner	Computational Biochemistry Research at ETH
155	D. Crippa	A Special Case of the Dynamization Problem for Least Cost Paths
156	R. Griesemer C. Pfister (ed.), B. Heeb, J. Templ	On the Linearization of Graphs and Writing Symbol Files Oberon Technical Notes
157	T. Weibel, G. Gonnet	An Algebra of Properties
158	M. Scholl (ed.)	Grundlagen von Datenbanken (Kurfassungen des 3.GI-Workshops, Volkse, 21. - 24.5.91)
159	K. Gates	Using Inverse Iteration to Improve the Divide and Conquer Algorithm
160	H. Mössenböck	Differences between Oberon and Oberon-2 The programming Language Oberon-2
161	S. Lalis	XNet: Supporting Distributed Programming in the Oberon Environment
162	G. Weikum, C. Hasse	Multi-Level Transaction Management for Complex Objects: Implementation, Performance, Parallelism
163	J. Nievergelt et al.	e**X**perimental geomet**r**i**Y** **Z**urich: Software for Geometric Computation
164	G.H. Gonnet, H. Straub	The Dynamic Programming Algorithm as a Finite Automation