SIAM J. MATRIX ANAL. APPL.
Vol. 16, No. 1, pp. 172–191, January 1995

012

A DIVIDE-AND-CONQUER ALGORITHM FOR THE SYMMETRIC TRIDIAGONAL EIGENPROBLEM*

MING GU† AND STANLEY C. EISENSTAT‡

Abstract. The authors present a stable and efficient divide-and-conquer algorithm for computing the spectral decomposition of an $N \times N$ symmetric tridiagonal matrix. The key elements are a new, stable method for finding the spectral decomposition of a symmetric arrowhead matrix and a new implementation of deflation. Numerical results show that this algorithm is competitive with bisection with inverse iteration, Cuppen's divide-and-conquer algorithm, and the QR algorithm for solving the symmetric tridiagonal eigenproblem.

Key words. symmetric tridiagonal eigenproblem, divide-and-conquer, arrowhead matrix

AMS subject classification. 65F15

1. Introduction.

Given an $N \times N$ symmetric tridiagonal matrix

$$T = \begin{pmatrix} \alpha_1 & \beta_2 & & & \\ \beta_2 & \alpha_2 & \beta_3 & & \\ & \ddots & \ddots & \ddots & \\ & & \beta_{N-1} & \alpha_{N-1} & \beta_N \\ & & & \beta_N & \alpha_N \end{pmatrix},$$

the *symmetric tridiagonal eigenproblem* is to find the spectral decomposition $T = X\Lambda X^T$, where Λ is diagonal and X is orthogonal. The diagonal elements of Λ are the eigenvalues of T, and the columns of X are the corresponding eigenvectors. In this paper we propose an *arrowhead divide-and-conquer* algorithm (ADC) for solving this problem.

ADC divides[1] T into two smaller symmetric tridiagonal matrices T_1 and T_2, each of which is a principle submatrix of T. It then recursively computes the spectral decompositions of T_1 and T_2 and constructs an orthogonal matrix Q such that $T = QHQ^T$, where

$$H = \begin{pmatrix} \alpha & z^T \\ z & D \end{pmatrix},$$

with D a diagonal matrix and z a vector, is a *symmetric arrowhead matrix*. Finally it finds the eigenvalues of T by computing the spectral decomposition $H = U\Lambda U^T$, where U is an orthogonal matrix, and computes the eigenvector matrix of T as QU.

Since error is associated with computation, a *numerical spectral decomposition* of T or H is usually defined as a decomposition of the form

$$T = \hat{X}\hat{\Lambda}\hat{X}^T + O(\epsilon\,\|T\|_2) \quad \text{or} \quad H = \hat{U}\hat{\Lambda}\hat{U}^T + O(\epsilon\,\|H\|_2), \tag{1}$$

* Received by the editors December 7, 1992; accepted for publication (in revised form) by J. R. Bunch, September 1, 1993. This research was supported in part by U. S. Army Research Office contract DAAL03-91-G-0032.

† Department of Mathematics and Lawrence Berkeley Laboratory, University of California, Berkeley, California 94720 (minggu@math.berkeley.edu).

‡ Department of Computer Science, Yale University, Box 208285, New Haven, Connecticut 06520-8285 (eisenstat-stan@cs.yale.edu).

[1] This strategy has previously appeared in [1], [3], [13], [15], [19], and [22].

where ϵ is the machine precision, $\hat{\Lambda}$ is diagonal, and $\hat{X}$ or $\hat{U}$ is *numerically orthogonal*. An algorithm that produces such a decomposition is said to be *stable*.

While the eigenvalues of T and H are always well conditioned with respect to a symmetric perturbation, the eigenvectors can be extremely sensitive to such perturbations [14, pp. 413–414]. That is, $\hat{\Lambda}$ must be close to Λ, but $\hat{X}$ and $\hat{U}$ can be very different from X and U, respectively. Thus one is usually content with stable algorithms for computing the spectral decompositions of T and H.

Finding the spectral decomposition of a symmetric arrowhead matrix is an interesting problem in its own right (see [3], [4], [26]–[28] and references therein). Several methods for solving this problem have been proposed [3], [15], [26], [28]. While they can compute the eigenvalues to high absolute accuracy, in the presence of close eigenvalues they can have difficulties in computing numerically orthogonal eigenvectors, unless extended precision arithmetic is used [24], [29]. In this paper we present a new algorithm for computing the spectral decomposition of a symmetric arrowhead matrix. It is similar to previous methods for finding the eigenvalues, but it uses a completely different approach to finding the eigenvectors, one that is stable. The amount of work is roughly the same as for previous methods, yet it does not require the use or simulation of extended precision arithmetic. Since it uses this algorithm, ADC is stable as well.

ADC computes all the eigenvalues of T in $O(N^2)$ time and both the eigenvalues and eigenvectors of T in $O(N^3)$ time. We show that by using the fast multipole method of Carrier, Greengard, and Rokhlin [10], [16], ADC can be accelerated to compute all the eigenvalues in $O(N \log_2 N)$ time and both the eigenvalues and eigenvectors in $O(N^2)$ time. These asymptotic time requirements are better than the corresponding worst-case times ($O(N^2)$ and $O(N^3)$) for bisection with inverse iteration [21], [23] and the QR algorithm [8]. Our algorithm for finding all the eigenvalues of H takes $O(N^2)$ time as do previous methods [3], [15], [26], [28]. By using the fast multipole method, it can be accelerated to compute all the eigenvalues in $O(N)$ time.

Cuppen's divide-and-conquer algorithm (CDC) [11], [12] uses a similar dividing strategy, but it reduces T to a symmetric rank-one modification to a diagonal matrix rather than to a symmetric arrowhead matrix. However, in the presence of close eigenvalues it can have difficulties in computing numerically orthogonal eigenvectors [11], [12], unless extended precision arithmetic is used [5], [24], [29]. In contrast, ADC is stable and is roughly twice as fast as existing implementations of CDC (e.g., `TREEQL` [12]) for large matrices due to the differences in how deflation is implemented.[2] ADC is also very competitive with bisection with inverse iteration [21], [23] and the QR algorithm [8].

Section 2 presents the dividing strategy used in ADC; §3 develops an efficient algorithm for the spectral decomposition of a symmetric arrowhead matrix and shows that it is stable; §4 discusses the deflation procedure used in ADC; §5 discusses the application of the fast multipole method to speed up ADC; and §6 presents some numerical results.

[2] Our techniques [17], [20] can be used to stabilize CDC without the need for extended precision arithmetic; our deflation procedure can be adapted to CDC, as can the fast multipole method.

We take the usual model of arithmetic[3]

$$\mathrm{fl}(x \circ y) = (x \circ y)\,(1+\xi),$$

where x and y are floating-point numbers; $\circ$ is one of $+, -, \times$, and $\div$; $\mathrm{fl}(x \circ y)$ is the floating-point result of the operation $\circ$; and $|\xi| \leq \epsilon$. We also require that

$$\mathrm{fl}(\sqrt{x}) = \sqrt{x}\,(1+\xi)$$

for any positive floating-point number x. For simplicity we ignore the possibility of overflow and underflow.

2. "Dividing" the matrix. Given an $N \times N$ symmetric tridiagonal matrix T, ADC divides T into two subproblems as follows:

$$T = \begin{pmatrix} T_1 & \beta_{k+1}e_k & 0 \\ \beta_{k+1}e_k^T & \alpha_{k+1} & \beta_{k+2}e_1^T \\ 0 & \beta_{k+2}e_1 & T_2 \end{pmatrix}, \tag{2}$$

where $1 < k < n$, T_1 and T_2 are $k \times k$ and $(N-k-1) \times (N-k-1)$ principle submatrices of T, respectively, and e_j is the jth unit vector of appropriate dimension. Usually k is taken to be $\lfloor N/2 \rfloor$.

Let $Q_i D_i {Q_i}^T$ be a spectral decomposition of T_i. Substituting into (2), we get

$$\begin{aligned} T &= \begin{pmatrix} Q_1 D_1 {Q_1}^T & \beta_{k+1}e_k & 0 \\ \beta_{k+1}e_k^T & \alpha_{k+1} & \beta_{k+2}e_1^T \\ 0 & \beta_{k+2}e_1 & Q_2 D_2 {Q_2}^T \end{pmatrix} \\ &= \begin{pmatrix} 0 & Q_1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & Q_2 \end{pmatrix} \begin{pmatrix} \alpha_{k+1} & \beta_{k+1}l_1^T & \beta_{k+2}f_2^T \\ \beta_{k+1}l_1 & D_1 & 0 \\ \beta_{k+2}f_2 & 0 & D_2 \end{pmatrix} \begin{pmatrix} 0 & 1 & 0 \\ {Q_1}^T & 0 & 0 \\ 0 & 0 & {Q_2}^T \end{pmatrix} \\ &\equiv QHQ^T, \end{aligned} \tag{3}$$

where l_1^T is the last row of Q_1 and f_2^T is the first row of Q_2. Thus T is reduced to the symmetric arrowhead matrix H by the orthogonal similarity transformation Q.

ADC computes the spectral decomposition $H = U\Lambda U^T$ using the algorithm described in §3. The eigenvalues of T are the diagonal elements of Λ, and the eigenvector matrix of T is obtained by computing the matrix-matrix product $X = QU$. To compute the spectral decompositions of T_1 and T_2, this process ((2) and (3)) can be recursively applied until the subproblems are sufficiently small. These small subproblems are solved using the QR algorithm. There can be at most $O(\log_2 N)$ levels of recursion.

Equations (2) and (3) also suggest a recursion for computing only the eigenvalues. Let f_1^T be the first row of Q_1 and let l_2^T be the last row of Q_2. Suppose that D_i, f_i, and l_i are given for $i = 1, 2$. Then after finding the spectral decomposition of H, the first row of X can be computed as $(0,\ f_1^T,\ 0)\ U$ and the last row of X can be computed as $(0,\ 0,\ l_2^T)\ U$. There is a similar recursion for CDC [11].

[3] This model excludes machines like CRAY and CDC Cyber that do not have a guard digit. ADC can easily be modified for such machines.

3. Computing the spectral decomposition of a symmetric arrowhead matrix. In this section we develop a stable and efficient method for finding the spectral decomposition of an $n \times n$ symmetric arrowhead matrix

$$H = \begin{pmatrix} \alpha & z^T \\ z & D \end{pmatrix},$$

where $D = \operatorname{diag}(d_2, \ldots, d_n)$ is an $(n-1) \times (n-1)$ matrix with $d_2 \leq d_3 \leq \cdots \leq d_n$, $z = (z_2, \ldots, z_n)^T$ is a vector of length $n-1$, and α is a scalar. The development closely parallels that in [17] and [20] for finding the spectral decomposition of a symmetric rank-one modification to a diagonal matrix.

We further assume that

$$d_{j+1} - d_j \geq \tau \|H\|_2 \quad \text{and} \quad |z_i| \geq \tau \|H\|_2, \tag{4}$$

where τ is a small multiple of ϵ to be specified later. Any symmetric arrowhead matrix can be reduced to one that satisfies these conditions by using the deflation procedure described in §4.1 and a simple permutation.

The following lemma characterizes the eigenvalues and eigenvectors of symmetric arrowhead matrices.

LEMMA 3.1 (Wilkinson [30, pp. 95–96], O'Leary and Stewart [26]). *The eigenvalues* $\{\lambda_i\}_{i=1}^n$ *of* H *satisfy the* interlacing property

$$\lambda_1 < d_2 < \lambda_2 < \cdots < d_n < \lambda_n$$

and the secular equation

$$f(\lambda) = \lambda - \alpha + \sum_{j=2}^{n} \frac{z_j^2}{d_j - \lambda} = 0.$$

For each eigenvalue λ_i *of* H, *the corresponding eigenvector is given by*

$$u_i = \left(-1, \frac{z_1}{d_2 - \lambda_i}, \ldots, \frac{z_n}{d_n - \lambda_i}\right)^T \Bigg/ \sqrt{1 + \sum_{j=2}^{n} \frac{z_j^2}{(d_j - \lambda_i)^2}}\,. \tag{5}$$

The following lemma allows us to construct a symmetric arrowhead matrix from its eigenvalues and its shaft.

LEMMA 3.2 (Boley and Golub [6]). *Given a set of numbers* $\{\hat{\lambda}_i\}_{i=1}^n$ *and a diagonal matrix* $D = \operatorname{diag}(d_2, \ldots, d_n)$ *satisfying the interlacing property*

$$\hat{\lambda}_1 < d_2 < \hat{\lambda}_2 < \cdots < d_n < \hat{\lambda}_n, \tag{6}$$

there exists a symmetric arrowhead matrix

$$\hat{H} = \begin{pmatrix} \hat{\alpha} & \hat{z}^T \\ \hat{z} & D \end{pmatrix}$$

whose eigenvalues are $\{\hat{\lambda}_i\}_{i=1}^n$. *The vector* $\hat{z} = (\hat{z}_2, \ldots, \hat{z}_n)^T$ *and the scalar* $\hat{\alpha}$ *are given by*

$$|\hat{z}_i| = \sqrt{(d_i - \hat{\lambda}_1)\,(\hat{\lambda}_n - d_i) \prod_{j=2}^{i-1} \frac{(\hat{\lambda}_j - d_i)}{(d_j - d_i)} \prod_{j=i}^{n-1} \frac{(\hat{\lambda}_j - d_i)}{(d_{j+1} - d_i)}} \tag{7}$$

$$\hat{\alpha} = \hat{\lambda}_1 + \sum_{j=2}^{n} \left(\hat{\lambda}_j - d_j\right), \tag{8}$$

where the sign of $\hat{z}_i$ *can be chosen arbitrarily.*

3.1. Computing the eigenvectors. If λ_i were given *exactly*, then we could compute each difference, each ratio, each product, and each sum in (5) to high relative accuracy, and thus compute u_i to componentwise high relative accuracy. In practice we can only hope to compute an approximation $\hat{\lambda}_i$ to λ_i. But problems can arise if we approximate u_i by

$$\hat{u}_i = \left(-1, \frac{z_1}{d_2 - \hat{\lambda}_i}, \ldots, \frac{z_n}{d_n - \hat{\lambda}_i}\right)^T \Bigg/ \sqrt{1 + \sum_{j=2}^{n} \frac{z_j^2}{(d_j - \hat{\lambda}_i)^2}}$$

(i.e., replace λ_i by $\hat{\lambda}_i$ in (5), as in [3], [15], and [26]). For even if $\hat{\lambda}_i$ is close to λ_i, the approximate ratio $z_j/(d_j - \hat{\lambda}_i)$ can be very different from the exact ratio $z_j/(d_j - \lambda_i)$, resulting in a $\hat{u}_i$ very different from u_i. And when all the approximate eigenvalues $\{\hat{\lambda}_i\}_{i=1}^n$ are computed and all the corresponding eigenvectors are approximated in this manner, the resulting eigenvector matrix may not be numerically orthogonal.

Lemma 3.2 allows us to overcome this problem. After we have computed all the approximate eigenvalues $\{\hat{\lambda}_i\}_{i=1}^n$ of H, we can find a *new* matrix $\hat{H}$ whose *exact* eigenvalues are $\{\hat{\lambda}_i\}_{i=1}^n$, and then compute the eigenvectors of $\hat{H}$ using Lemma 3.1. Note that each difference, each product, and each ratio in (7) can be computed to high relative accuracy, and the sign of $\hat{z}_i$ can be taken to be the sign of z_i. Thus $\hat{z}_i$ can be computed to componentwise high relative accuracy. Substituting the *exact* eigenvalues $\{\hat{\lambda}_i\}_{i=1}^n$ and the computed $\hat{z}$ into (5), each eigenvector of $\hat{H}$ can be computed to componentwise high relative accuracy. And, after all the eigenvectors are computed, the computed eigenvector matrix of $\hat{H}$ will be numerically orthogonal.

To ensure the existence of $\hat{H}$, the approximations $\{\hat{\lambda}_i\}_{i=1}^n$ must satisfy the interlacing property (6). But since the exact eigenvalues of H satisfy the same interlacing property (see Lemma 3.1), this is only an accuracy requirement on the computed eigenvalues and is not an additional restriction on H.

We can use the spectral decomposition of $\hat{H}$ as an approximation to that of H. Since

$$H = \begin{pmatrix} \alpha & z^T \\ z & D \end{pmatrix} = \hat{H} + \begin{pmatrix} \alpha - \hat{\alpha} & (z - \hat{z})^T \\ z - \hat{z} & 0 \end{pmatrix},$$

we have

$$\|\hat{H} - H\|_2 \leq |\alpha - \hat{\alpha}| + \|z - \hat{z}\|_2.$$

Thus such a substitution is stable (see (1)) as long as $\hat{\alpha}$ and $\hat{z}$ are close to α and z, respectively (cf. [17], [20]).

3.2. Computing the eigenvalues. To guarantee that $\hat{z}$ is close to z and $\hat{\alpha}$ is close to α, we must ensure that $\{\hat{\lambda}_i\}_{i=1}^n$ are sufficiently accurate approximations to the eigenvalues. The key is the stopping criterion for the root-finder, which requires a slight reformulation of the secular equation (cf. [9], [17], [20]).

Consider the eigenvalue $\lambda_i \in (d_i, d_{i+1})$, where $2 \le i \le n-1$; we consider the cases $i = 1$ and $i = n$ later. λ_i is a root of the secular equation

$$f(\lambda) = \lambda - \alpha + \sum_{j=2}^{n} \frac{z_j^2}{d_j - \lambda} = 0.$$

We first assume that[4] $\lambda_i \in \left(d_i, \frac{d_i+d_{i+1}}{2}\right)$. Let $\alpha_i = d_i - \alpha$ and $\delta_j = d_j - d_i$, and let

$$\psi_i(\mu) \equiv \sum_{j=2}^{i} \frac{z_j^2}{\delta_j - \mu} \quad \text{and} \quad \phi_i(\mu) \equiv \sum_{j=i+1}^{n} \frac{z_j^2}{\delta_j - \mu}.$$

Since

$$f(\mu + d_i) = \mu + \alpha_i + \psi_i(\mu) + \phi_i(\mu) \equiv g_i(\mu),$$

we seek the root $\mu_i = \lambda_i - d_i \in (0, \delta_{i+1}/2)$ of $g_i(\mu) = 0$. Let $\hat{\mu}_i$ be the computed root so that $\hat{\lambda}_i = d_i + \hat{\mu}_i$ is the computed eigenvalue.

An important property of $g_i(\mu)$ is that each difference $\delta_j - \mu$ can be evaluated to high relative accuracy for any $\mu \in (0, \delta_{i+1}/2)$. Indeed, since $\delta_i = 0$, we have $\mathrm{fl}(\delta_i - \mu) = -\mathrm{fl}(\mu)$. Since $\mathrm{fl}(\delta_{i+1}) = \mathrm{fl}(d_{i+1} - d_i)$ and $0 < \mu < (d_{i+1} - d_i)/2$, we can compute $\mathrm{fl}(\delta_{i+1} - \mu)$ as $\mathrm{fl}(\mathrm{fl}(d_{i+1} - d_i) - \mathrm{fl}(\mu))$. In a similar fashion, we can compute $\delta_j - \mu$ to high relative accuracy for any $j \neq i, i+1$.

Because of this property, each ratio $z_j^2/(\delta_j - \mu)$ in $g_i(\mu)$ can be evaluated to high relative accuracy for any $\mu \in (0, \delta_{i+1}/2)$. Moreover, α_i can be computed to high relative accuracy. Thus, since both $\psi_i(\mu)$ and $\phi_i(\mu)$ are sums of terms of the same sign, we can bound the error in computing $g_i(\mu)$ by

$$\eta n(|\mu| + |\alpha_i| + |\psi_i(\mu)| + |\phi_i(\mu)|),$$

where η is a small multiple of ϵ that is independent of n and μ.

We now assume that $\lambda_i \in \left[(d_i + d_{i+1})/2, d_{i+1}\right)$. Let $\alpha_i = d_{i+1} - \alpha$ and $\delta_j = d_j - d_{i+1}$, and let

$$\psi_i(\mu) \equiv \sum_{j=2}^{i} \frac{z_j^2}{\delta_j - \mu} \quad \text{and} \quad \phi_i(\mu) \equiv \sum_{j=i+1}^{n} \frac{z_j^2}{\delta_j - \mu}.$$

We seek the root $\mu_i = \lambda_i - d_{i+1} \in [\delta_i/2, 0)$ of the equation

$$g_i(\mu) \equiv f(\mu + d_{i+1}) = \mu + \alpha_i + \psi_i(\mu) + \phi_i(\mu) = 0.$$

[4] This can be checked by computing $f\left(\frac{d_i+d_{i+1}}{2}\right)$. If $f\left(\frac{d_i+d_{i+1}}{2}\right) > 0$, then $\lambda_i \in \left(d_i, \frac{d_i+d_{i+1}}{2}\right)$, otherwise $\lambda_i \in \left[\frac{d_i+d_{i+1}}{2}, d_{i+1}\right)$.

Let $\hat{\mu}_i$ be the computed root so that $\hat{\lambda}_i = d_{i+1} + \hat{\mu}_i$. For any $\mu \in [\delta_i/2, 0)$, each difference $\delta_j - \mu$ can again be computed to high relative accuracy, as can each ratio $z_j^2/(\delta_j - \mu)$ and the scalar α_i, and we can bound the error in computing $g_i(\mu)$ as before.

Next we consider the case $i = 1$. Let $\alpha_1 = d_2 - \alpha$ and $\delta_j = d_j - d_2$, and let

$$\psi_1(\mu) \equiv 0 \quad \text{and} \quad \phi_1(\mu) \equiv \sum_{j=2}^{n} \frac{z_j^2}{\delta_j - \mu}.$$

We seek the root $\mu_1 = \lambda_1 - d_2 \in (-\|z\|_2 - |\alpha_1|, 0)$ of the equation

$$g_1(\mu) \equiv f(\mu + d_2) = \mu + \alpha_1 + \psi_1(\mu) + \phi_1(\mu) = 0.$$

Let $\hat{\mu}_1$ be the computed root so that $\hat{\lambda}_1 = d_2 + \hat{\mu}_1$. For any $\mu \in (-\|z\|_2 - |\alpha_1|, 0)$, each ratio $z_j^2/(\delta_j - \mu)$ can be computed to high relative accuracy, as can α_1, and we can bound the error in computing $g_1(\mu)$ as before.

Finally we consider the case $i = n$. Let $\alpha_n = d_n - \alpha$ and $\delta_j = d_j - d_n$, and let

$$\psi_n(\mu) \equiv \sum_{j=2}^{n} \frac{z_j^2}{\delta_j - \mu} \quad \text{and} \quad \phi_n(\mu) \equiv 0.$$

We seek the root $\mu_n = \lambda_n - d_n \in (0, \|z\|_2 + |\alpha_n|)$ of the equation

$$g_n(\mu) \equiv f(\mu + d_n) = \mu + \alpha_n + \psi_n(\mu) + \phi_n(\mu) = 0.$$

Let $\hat{\mu}_n$ be the computed root so that $\hat{\lambda}_n = d_n + \hat{\mu}_n$. For any $\mu \in (0, \|z\|_2 + |\alpha_n|)$, each ratio $z_j^2/(\delta_j - \mu)$ can be computed to high relative accuracy, as can α_n, and we can bound the error in computing $g_n(\mu)$ as before.

In practice the root-finder cannot make any progress at a point μ where it is impossible to determine the sign of $g_i(\mu)$ numerically. Thus we propose the stopping criterion

$$|g_i(\mu)| \leq \eta n \left(|\mu| + |\alpha_i| + |\psi_i(\mu)| + |\phi_i(\mu)| \right), \tag{9}$$

where, as before, the right-hand side is an upper bound on the round-off error in computing $g_i(\mu)$. Note that for each i, there is at least one floating-point number that satisfies this stopping criterion numerically, namely, $\mathrm{fl}(\mu_i)$.

We have not specified the method used to find the root of $g_i(\mu)$. We used a modified version of the rational interpolation strategy in [9] for the numerical experiments, but bisection and its variations [26], [28] or the improved rational interpolation strategies in [15], [25] would also work. What is most important is the stopping criterion and the fact that, with the reformulation of the secular equation given above, we can find a μ that satisfies it.

3.3. Numerical stability. In this subsection we show that $\hat{\alpha}$ and $\hat{z}$ are indeed close to α and z, respectively, as long as the root-finder guarantees that each $\hat{\mu}_i$ satisfies the stopping criterion (9).

Since $f(\lambda_i) = 0$, we have

$$|\alpha_i| = \left| -\mu_i - \sum_{j=2}^{n} \frac{z_j^2}{d_j - \lambda_i} \right| \leq |\mu_i| + \sum_{j=2}^{n} \frac{z_j^2}{|d_j - \lambda_i|}$$

and

$$f(\hat{\lambda}_i) = f(\hat{\lambda}_i) - f(\lambda_i) = (\hat{\lambda}_i - \lambda_i)\left(1 + \sum_{j=2}^{n} \frac{z_j^2}{(d_j - \hat{\lambda}_i)(d_j - \lambda_i)}\right).$$

Since the computed eigenvalue $\hat{\lambda}_i$ satisfies (9), we have

$$|f(\hat{\lambda}_i)| \le \eta n \left(|\mu_i| + |\hat{\mu}_i| + \sum_{j=2}^{n} \frac{z_j^2}{|d_j - \lambda_i|} + \sum_{j=2}^{n} \frac{z_j^2}{|d_j - \hat{\lambda}_i|}\right),$$

so that

$$|\hat{\lambda}_i - \lambda_i| \left(1 + \sum_{j=2}^{n} \frac{z_j^2}{|(d_j - \hat{\lambda}_i)(d_j - \lambda_i)|}\right) \le \eta n \left(|\mu_i| + |\hat{\mu}_i| + \sum_{j=2}^{n} \frac{z_j^2}{|d_j - \hat{\lambda}_i|} + \sum_{j=2}^{n} \frac{z_j^2}{|d_j - \lambda_i|}\right). \tag{10}$$

Note that for any i and j,

$$|\mu_i| + |\hat{\mu}_i| \le 4\|H\|_2 + |\hat{\lambda}_i - \lambda_i| \quad \text{and} \quad |d_j - \hat{\lambda}_i| + |d_j - \lambda_i| \le 4\|H\|_2 + |\hat{\lambda}_i - \lambda_i|.$$

Substituting these relations into (10), we get

$$|\hat{\lambda}_i - \lambda_i| \left(1 + \sum_{j=2}^{n} \frac{z_j^2}{|(d_j - \hat{\lambda}_i)(d_j - \lambda_i)|}\right) \le \eta n \left(4\|H\|_2 + |\hat{\lambda}_i - \lambda_i|\right) \left(1 + \sum_{j=2}^{n} \frac{z_j^2}{|d_j - \hat{\lambda}_i||d_j - \lambda_i|}\right),$$

or

$$|\hat{\lambda}_i - \lambda_i| \le \frac{4\eta n \|H\|_2}{1 - \eta n},$$

i.e., all the eigenvalues are computed to high absolute accuracy. Applying (8) in Lemma 3.2 to both H and $\hat{H}$, we have

$$\alpha = \lambda_1 + \sum_{j=2}^{n} (\lambda_j - d_j) \quad \text{and} \quad \hat{\alpha} = \hat{\lambda}_1 + \sum_{j=2}^{n} (\hat{\lambda}_j - d_j),$$

and therefore

$$|\alpha - \hat{\alpha}| = \left|\sum_{j=1}^{n} (\lambda_j - \hat{\lambda}_j)\right| \le \sum_{j=1}^{n} \left|\lambda_j - \hat{\lambda}_j\right| \le \frac{4\eta n^2 \|H\|_2}{1 - \eta n}. \tag{11}$$

To show that $\hat{z}$ is close to z, we further note that for any i and j, we have

$$|\hat{\mu}_i| \leq |\mu_i| + |\hat{\lambda}_i - \lambda_i|$$

and

$$\frac{1}{|d_j - \hat{\lambda}_i|} + \frac{1}{|d_j - \lambda_i|} \leq \frac{2}{|(d_j - \hat{\lambda}_i)(d_j - \lambda_i)|^{\frac{1}{2}}} + \frac{|\hat{\lambda}_i - \lambda_i|}{|(d_j - \hat{\lambda}_i)(d_j - \lambda_i)|}.$$

Substituting these relations into (10) and using the Cauchy–Schwartz inequality, we get

$$\begin{aligned}
|\hat{\lambda}_i - \lambda_i| & \left(1 + \sum_{j=2}^{n} \frac{z_j^2}{|(d_j - \hat{\lambda}_i)(d_j - \lambda_i)|}\right) \\
& \leq \frac{2\eta n}{1 - \eta n} \left(|\mu_i| + \sum_{j=2}^{n} \frac{z_j^2}{|(d_j - \hat{\lambda}_i)(d_j - \lambda_i)|^{1/2}}\right) \\
& \leq \frac{2\eta n}{1 - \eta n} \sqrt{|\mu_i|^2 + \|z\|_2^2} \sqrt{1 + \sum_{j=2}^{n} \frac{z_j^2}{|(d_j - \hat{\lambda}_i)(d_j - \lambda_i)|}}.
\end{aligned}$$

Since $|\mu_i|^2 + \|z\|_2^2 \leq 5\|H\|_2^2$, we have

$$\begin{aligned}
|\hat{\lambda}_i - \lambda_i| & \leq \frac{2\eta n}{1 - \eta n} \sqrt{|\mu_i|^2 + \|z\|_2^2} \Bigg/ \sqrt{1 + \sum_{j=2}^{n} \frac{z_j^2}{|(d_j - \hat{\lambda}_i)(d_j - \lambda_i)|}} \\
& \leq \frac{2\sqrt{5}\eta n \|H\|_2}{(1 - \eta n)|z_j|} \sqrt{|(d_j - \hat{\lambda}_i)(d_j - \lambda_i)|} \\
& \leq \frac{2\sqrt{5}\eta n \|H\|_2}{(1 - \eta n)|z_j|} \left(|d_j - \lambda_i| + \frac{1}{2}|\hat{\lambda}_i - \lambda_i|\right).
\end{aligned}$$

Letting $\beta_j = 2\sqrt{5}\eta n \|H\|_2 / ((1 - \eta n)|z_j|)$, this implies that

$$|\hat{\lambda}_i - \lambda_i| \leq \frac{\beta_j}{1 - \frac{1}{2}\beta_j} |d_j - \lambda_i| \tag{12}$$

for every $2 \leq j \leq n$, provided that $\beta_j < 2$.

Let $\hat{\lambda}_i - \lambda_i = \alpha_{ij}(d_j - \lambda_i)/z_j$ for all i and j. Suppose that we pick $\tau = 6\eta n^2$ in (4). Then $|z_j| \geq 6\eta n^2 \|H\|_2$. Assume further that $\eta n < 1/100$. Then $\beta_j \leq 2/5$, and (12) implies that $|\alpha_{ij}| \leq \alpha \equiv 6\eta n \|H\|_2$ for all i and j. Thus

$$|\hat{z}_i| = \sqrt{\frac{\prod_{j=1}^{n} (\hat{\lambda}_j - d_i)}{\prod_{j=2, j\neq i}^{n} (d_j - d_i)}} = \sqrt{\frac{\prod_{j=1}^{n} (\lambda_j - d_i)\left(1 + \frac{\alpha_{ji}}{z_i}\right)}{\prod_{j=2, j\neq i}^{n} (d_j - d_i)}} = |z_i| \sqrt{\prod_{j=1}^{n} \left(1 + \frac{\alpha_{ji}}{z_i}\right)}$$

and, since $\hat{z}_i$ and z_i have the same sign,

$$|\hat{z}_i - z_i| = |z_i| \left|\sqrt{\prod_{j=1}^{n} \left(1 + \frac{\alpha_{ji}}{z_i}\right)} - 1\right| \leq |z_i| \left(\left(1 + \frac{\alpha}{|z_i|}\right)^{\frac{n}{2}} - 1\right)$$

$$\begin{aligned}(13)\qquad &\le |z_i| \left(\exp\left(\frac{\alpha n}{2|z_i|}\right) - 1\right) \le (e-1)\,\alpha n/2 \\ &\le 6\eta n^2 \|H\|_2,\end{aligned}$$

where we have used the fact that $\alpha n/(2|z_i|) \le 1$ and that $(e^x - 1)/x \le e - 1$ for $0 < x \le 1$.

One factor of n in τ and the bounds (11) and (13) comes from the stopping criterion (9). This is quite conservative and could be reduced to $\log_2 n$ by using a binary tree structure for summing the terms in $\psi_i(\mu)$ and $\phi_i(\mu)$. The other factor of n comes from the upper bound for $\sum_{j=1}^{n}(\lambda_j - \hat{\lambda}_j)$ in (11) and $\prod_{j=1}^{n}(1 + \alpha_{ji}/z_i)$ in (13). This also seems quite conservative. Thus we might expect the factor of n^2 in τ and the bounds (11) and (13) to be more like $O(n)$ in practice.

4. Deflation.

4.1. Deflation for symmetric arrowhead matrices.

Let

$$H = \begin{pmatrix} \alpha & z^T \\ z & D \end{pmatrix},$$

where $D = \operatorname{diag}(d_2, \ldots, d_n)$ and $z = (z_2, \ldots, z_n)^T$. We now show that we can reduce[5] H to a symmetric arrowhead matrix that further satisfies

$$|d_i - d_j| \ge \tau\|H\|_2, \quad \text{for} \quad i \ne j \quad \text{and} \quad |z_i| \ge \tau\|H\|_2$$

(cf. (4)), where τ is specified in §3.3. We illustrate the reductions for $n = 3$, $i = 3$, and $j = 2$.

Assume that $|z_i| < \tau\|H\|_2$. Then

$$(14)\qquad H = \begin{pmatrix} \alpha & z_2 & z_3 \\ z_2 & d_2 & \\ z_3 & & d_3 \end{pmatrix} = \begin{pmatrix} \alpha & z_2 & 0 \\ z_2 & d_2 & \\ 0 & & d_3 \end{pmatrix} + O(\tau\|H\|_2).$$

We perturb z_i to zero. Then H is perturbed by $O(\tau\|H\|_2)$. d_i is an eigenvalue of the perturbed matrix and is deflated. The $(n-1) \times (n-1)$ leading principle submatrix of the perturbed matrix is another symmetric arrowhead matrix with smaller dimensions. This deflation rule is stable (see (1)).

Now assume that $|d_i - d_j| < \tau\|H\|_2$. Apply a Givens rotation G to H to zero out z_i:

$$\begin{aligned}GHG^T &= \begin{pmatrix} 1 & & \\ & c & s \\ & -s & c \end{pmatrix} \begin{pmatrix} \alpha & z_2 & z_3 \\ z_2 & d_2 & \\ z_3 & & d_3 \end{pmatrix} \begin{pmatrix} 1 & & \\ & c & -s \\ & s & c \end{pmatrix} \\ (15)\qquad &= \begin{pmatrix} \alpha & r & 0 \\ r & d_2c^2 + d_3s^2 & cs(d_3 - d_2) \\ 0 & cs(d_3 - d_2) & d_2s^2 + d_3c^2 \end{pmatrix} \\ &= \begin{pmatrix} \alpha & r & 0 \\ r & d_2c^2 + d_3s^2 & \\ 0 & & d_2s^2 + d_3c^2 \end{pmatrix} + O(\tau\|H\|_2),\end{aligned}$$

[5] These rules have previously appeared in [15] and [19].

where $r = \sqrt{z_i^2 + z_j^2}$, $c = z_j/r$, and $s = z_i/r$. We perturb $cs(d_i - d_j)$ to zero. Then GHG^T is perturbed by $O(\tau\|H\|_2)$. $d_j s^2 + d_i c^2$ is an eigenvalue of the perturbed matrix and can be deflated. The $(n-1) \times (n-1)$ leading principle submatrix of the perturbed matrix is another symmetric arrowhead matrix with smaller dimensions. This deflation rule is also stable (see (1)).

4.2. Local deflation. In the dividing strategy for ADC (see (3)), we write

$$T = \begin{pmatrix} 0 & Q_1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & Q_2 \end{pmatrix} \begin{pmatrix} \alpha_{k+1} & \beta_{k+1} l_1^T & \beta_{k+2} f_2^T \\ \beta_{k+1} l_1 & D_1 & 0 \\ \beta_{k+2} f_2 & 0 & D_2 \end{pmatrix} \begin{pmatrix} 0 & 1 & 0 \\ {Q_1}^T & 0 & 0 \\ 0 & 0 & {Q_2}^T \end{pmatrix} \tag{16}$$
$$= (QU)\Lambda(QU)^T,$$

where Q is the first matrix in (16) and $U\Lambda U^T$ is the spectral decomposition of the middle matrix.

Note that Q is a block matrix with some zero blocks. When we compute the matrix-matrix product QU, we would like to take advantage of this structure. Since the main cost of ADC is in computing such products, we get a speedup of close to a factor of two by doing so. This is not done in any current implementation of CDC.

If the vector $(\beta_{k+1} l_1^T,\ \beta_{k+2} f_2^T)$ has components with small absolute value, then we can apply reduction (14). The block structure of Q is preserved. If D_1 has two close diagonal elements, then we can apply reduction (15). The block structure of Q is again preserved. We can do the same when D_2 has two close diagonal elements.

However, when D_1 has a diagonal element that is close to a diagonal element in D_2 and we apply reduction (15), the block structure of Q is changed. To illustrate, assume that after applying a permutation the first diagonal element of D_1 is close to the last diagonal element of D_2. Let $Q_1 = (q_1, \tilde{Q}_1)$ and $Q_2 = (\tilde{Q}_2, q_2)$; let $D_1 = \mathrm{diag}(d_2, \tilde{D}_1)$ and $D_2 = \mathrm{diag}(\tilde{D}_2, d_N)$; and let $\beta_{k+1} l_1^T = (z_2, \tilde{z}_1^T)$ and $\beta_{k+2} f_2^T = (\tilde{z}_2^T, z_N)$. By assumption, d_2 and d_N are close. When we apply the Givens rotation

$$G = \begin{pmatrix} 1 & & & & \\ & c & & & s \\ & & & I_{N-3} & \\ & & & & \\ & -s & & & c \end{pmatrix}$$

to the middle matrix in (16) to zero out z_N, we create some nonzero elements in the second and Nth columns of Q:

$$T = \begin{pmatrix} 0 & q_1 & \tilde{Q}_1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \tilde{Q}_2 & q_2 \end{pmatrix} G^T G \begin{pmatrix} \alpha_{k+1} & z_2 & \tilde{z}_1^T & \tilde{z}_2^T & z_N \\ z_2 & d_2 & & & \\ \tilde{z}_1 & & \tilde{D}_1 & & \\ \tilde{z}_2 & & & \tilde{D}_2 & \\ z_N & & & & d_N \end{pmatrix} G^T G \begin{pmatrix} 0 & 1 & 0 \\ q_1^T & 0 & 0 \\ \tilde{Q}_1^T & 0 & 0 \\ 0 & 0 & \tilde{Q}_2^T \\ 0 & 0 & q_2^T \end{pmatrix}$$

$$= \begin{pmatrix} 0 & cq_1 & \tilde{Q}_1 & 0 & -sq_1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & sq_2 & 0 & \tilde{Q}_2 & cq_2 \end{pmatrix} \begin{pmatrix} \alpha_{k+1} & r & \tilde{z}_1^T & \tilde{z}_2^T & 0 \\ r & \tilde{d}_2 & & & \\ \tilde{z}_1 & & \tilde{D}_1 & & \\ \tilde{z}_2 & & & \tilde{D}_2 & \\ 0 & & & & \tilde{d}_N \end{pmatrix} \begin{pmatrix} 0 & 1 & 0 \\ cq_1^T & 0 & sq_2^T \\ \tilde{Q}_1^T & 0 & 0 \\ 0 & 0 & \tilde{Q}_2^T \\ -sq_2^T & 0 & cq_2^T \end{pmatrix}$$

$$+ O(\tau\|T\|_2),$$

where $\tilde{d}_2 = d_2c^2 + d_Ns^2$ and $\tilde{d}_N = d_2s^2 + d_Nc^2$.

Note that $\tilde{d}_N$ is an approximate eigenvalue of T and can be deflated. The corresponding approximate eigenvector is the last column of the first matrix. The leading $(N-1) \times (N-1)$ principle submatrix of the middle matrix is again a symmetric arrowhead matrix and can be deflated in a similar fashion until no diagonal element of $\tilde{D}_1$ is close to a diagonal element of $\tilde{D}_2$.

Thus, ignoring permutations of the columns of Q_i and the diagonal elements of D_i, after a series of these interblock deflations T can be written as

$$T = \begin{pmatrix} \tilde{X}_1 & \tilde{X}_2 \end{pmatrix} \begin{pmatrix} \tilde{H}_1 & \\ & \tilde{\Lambda}_2 \end{pmatrix} \begin{pmatrix} \tilde{X}_1 & \tilde{X}_2 \end{pmatrix}^T + O(\tau\|T\|_2). \tag{17}$$

$\tilde{\Lambda}_2$ is a diagonal matrix whose diagonal elements are the deflated eigenvalues and the columns of $\tilde{X}_2$ are the corresponding approximate eigenvectors. $\tilde{H}_1$ is the symmetric arrowhead matrix

$$\tilde{H}_1 = \begin{pmatrix} \alpha_{k+1} & \tilde{z}_0^T & \tilde{z}_1^T & \tilde{z}_2^T \\ \tilde{z}_0 & \tilde{D}_0 & & \\ \tilde{z}_1 & & \tilde{D}_1 & \\ \tilde{z}_2 & & & \tilde{D}_2 \end{pmatrix},$$

where the dimension of $\tilde{D}_0$ is the number of deflations, $\tilde{D}_1$ and $\tilde{D}_2$ contain the undeflated diagonal elements of D_1 and D_2, and $\tilde{z}_0$, $\tilde{z}_1$, and $\tilde{z}_2$ are defined accordingly. $\tilde{X}_1$ is of the form

$$\tilde{X}_1 = \begin{pmatrix} 0 & \tilde{Q}_{0,1} & \tilde{Q}_1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & \tilde{Q}_{0,2} & 0 & \tilde{Q}_2 \end{pmatrix}, \tag{18}$$

where the column dimension of both $\tilde{Q}_{0,1}$ and $\tilde{Q}_{0,2}$ is the number of deflations and the columns of $\tilde{Q}_1$ and $\tilde{Q}_2$ are those columns of Q_1 and Q_2 not affected by deflation.

If some diagonal element of $\tilde{D}_0$ is close to a diagonal element of either $\tilde{D}_1$ or $\tilde{D}_2$, then we can use reduction (15) to deflate without changing the structure of $\tilde{X}_1$. In the following we assume that no further such deflation is possible.

Let $\tilde{U}_1\tilde{\Lambda}_1\tilde{U}_1^T$ be the spectral decomposition of $\tilde{H}_1$. Then

$$\begin{aligned} T &= \begin{pmatrix} \tilde{X}_1 & \tilde{X}_2 \end{pmatrix} \begin{pmatrix} \tilde{U}_1\tilde{\Lambda}_1\tilde{U}_1^T & \\ & \tilde{\Lambda}_2 \end{pmatrix} \begin{pmatrix} \tilde{X}_1 & \tilde{X}_2 \end{pmatrix}^T + O(\tau\|T\|_2) \\ &= \begin{pmatrix} \tilde{X}_1\tilde{U}_1 & \tilde{X}_2 \end{pmatrix} \begin{pmatrix} \tilde{\Lambda}_1 & \\ & \tilde{\Lambda}_2 \end{pmatrix} \begin{pmatrix} \tilde{X}_1\tilde{U}_1 & \tilde{X}_2 \end{pmatrix}^T + O(\tau\|T\|_2). \end{aligned}$$

Thus $(\tilde{X}_1\tilde{U}_1, \tilde{X}_2)$ is an approximate eigenvector matrix of T. The matrix $\tilde{X}_1\tilde{U}_1$ can be computed while taking advantage of the block structure of $\tilde{X}_1$.

We refer to these deflations as *local* deflations since they are associated with individual subproblems of ADC.

4.3. Global deflation. To illustrate *global* deflation, we look at two levels of the dividing strategy (see (2)); for simplicity, we denote unimportant entries of T by x:

$$T = \begin{pmatrix} T_1 & \beta_{i+j+2}e_{i+j+1} & \\ \beta_{i+j+2}e_{i+j+1}^T & x & xe_1^T \\ & xe_1 & T_2 \end{pmatrix}$$
$$= \begin{pmatrix} T_{1,1} & xe_i & & & \\ xe_i^T & x & \beta_{i+2}e_1^T & & \\ & \beta_{i+2}e_1 & T_{1,2} & \beta_{i+j+2}e_j & \\ & & \beta_{i+j+2}e_j^T & x & xe_1^T \\ & & & xe_1 & T_2 \end{pmatrix},$$

where T_1, T_2, $T_{1,1}$, and $T_{1,2}$ are principle submatrices of T of dimensions $(i+j+1)\times(i+j+1)$, $(N-i-j-2)\times(N-i-j-2)$, $i\times i$, and $j\times j$, respectively.

Let $Q_{1,2}D_{1,2}Q_{1,2}^T$ be the spectral decomposition of $T_{1,2}$, and let $f_{1,2}^T$ and $l_{1,2}^T$ be the first and last rows of $Q_{1,2}$, respectively. Then

$$T = \begin{pmatrix} T_{1,1} & xe_i & & & \\ xe_i^T & x & \beta_{i+2}e_1^T & & \\ & \beta_{i+2}e_1 & Q_{1,2}D_{1,2}Q_{1,2}^T & \beta_{i+j+2}e_j & \\ & & \beta_{i+j+2}e_j^T & x & xe_1^T \\ & & & xe_1 & T_2 \end{pmatrix}$$
$$= Y \begin{pmatrix} T_{1,1} & xe_i & & & \\ xe_i^T & x & \beta_{i+2}f_{1,2}^T & & \\ & \beta_{i+2}f_{1,2} & D_{1,2} & \beta_{i+j+2}l_{1,2} & \\ & & \beta_{i+j+2}l_{1,2}^T & x & xe_1^T \\ & & & xe_1 & T_2 \end{pmatrix} Y^T, \tag{19}$$

where $Y = \mathrm{diag}(I_i,\ 1,\ Q_{1,2},\ 1,\ I_{N-i-j-2})$.

Let $\bar{d}_s$ be the sth diagonal element of $D_{1,2}$, and let $\bar{f}_s$ and $\bar{l}_s$ be the sth components of $f_{1,2}$ and $l_{1,2}$, respectively. Then, ignoring all zero components, the $(i+s+1)$st row of the middle matrix in (19) is $(\beta_{i+2}\bar{f}_s,\ \bar{d}_s,\ \beta_{i+j+2}\bar{l}_s)$. Thus if both $|\beta_{i+2}\bar{f}_s|$ and $|\beta_{i+j+2}\bar{l}_s|$ are small, then we can perturb them both to zero. $\bar{d}_s$ is an approximate eigenvalue of T and the $(i+s+1)$st column of Y is the corresponding approximate eigenvector. This eigenvalue and its eigenvector can be deflated from all subsequent subproblems. We call this *global* deflation.

Consider the deflation procedure for computing the spectral decomposition of T_1 in §4.2. If $|\beta_{i+2}\bar{f}_s|$ is small, then it can be perturbed to zero. This is a local deflation if only $|\beta_{i+2}\bar{f}_s|$ is small, and a global deflation if $|\beta_{i+j+2}\bar{l}_s|$ is also small.

5. Acceleration by the fast multipole method. Suppose that we want to evaluate the complex function

$$\Phi(\zeta) = \sum_{j=1}^{n} c_j \varphi(\zeta - \zeta_j) \tag{20}$$

at m points in the complex plane, where $\{c_j\}_{j=1}^n$ are constants and $\varphi(\zeta)$ is one of $\log(\zeta)$, $1/\zeta$, and $1/\zeta^2$. The direct computation takes $O(nm)$ time. But the *fast*

multipole method (FMM) of Carrier, Greengard, and Rokhlin [10], [16] takes only $O(n+m)$ time to approximate $\Phi(\zeta)$ at these points to a precision specified by the user.[6] In this section we briefly describe how FMM can be used to accelerate ADC. A more detailed description appears in [17] and [18] in the context of updating the singular value decomposition.

Let

$$H = \begin{pmatrix} \alpha & z^T \\ z & D \end{pmatrix},$$

where $D = \operatorname{diag}(d_2, \ldots, d_n)$ is an $(n-1) \times (n-1)$ matrix with $d_2 \le d_3 \le \cdots \le d_n$, $z = (z_2, \ldots, z_n)^T$ is a vector of length $n-1$, and α is a scalar. Let $U \Lambda U^T$ denote the spectral decomposition of H, with $U = (u_1, \ldots, u_n)$ and $\Lambda = \operatorname{diag}(\lambda_1, \ldots, \lambda_n)$.

Consider computing $U^T q$ for a vector $q = (q_1, \ldots, q_n)^T$. By (5) in Lemma 3.1, the ith component $u_i^T q$ of $U^T q$ can be written as

$$u_i^T q = \frac{-q_1 + \Phi_1(\lambda_i)}{\sqrt{1 + \Phi_2(\lambda_i)}},$$

where

$$\Phi_1(\lambda) = \sum_{k=2}^{n} \frac{z_k q_k}{d_k - \lambda} \quad \text{and} \quad \Phi_2(\lambda) = \sum_{k=2}^{n} \frac{z_k^2}{(d_k - \lambda)^2}.$$

Thus we can compute $U^T q$ by evaluating $\Phi_1(\lambda)$ and $\Phi_2(\lambda)$ at n points. Since these functions are of the form (20), we can do this in $O(n)$ time using FMM. To achieve better efficiency, we modify FMM to take advantage of the fact that all the computations are real (see [17]–[19]).

Let T be an $N \times N$ symmetric tridiagonal matrix. When ADC is used to compute all the eigenvalues and eigenvectors, the main cost for each subproblem is in forming $\tilde{X}_1 U$ (see (17)), where $\tilde{X}_1$ is a column orthogonal matrix.[7] Each row of $\tilde{X}_1 U$ is of the form $q^T U = (U^T q)^T$ and there are $O(n)$ rows. Thus the cost of computing $\tilde{X}_1 U$ is $O(n^2)$ using FMM. There are $\log_2 N$ levels of recursion and 2^{k-1} subproblems at the kth level, each of size $O(N/2^k)$. Thus the cost at the kth level is $O(N^2/2^k)$ and the total time is $O(N^2)$.

We may also have to apply the eigenvector matrix of T to an orthogonal matrix Y, e.g., when T is obtained by reducing a dense matrix to tridiagonal form [14, pp. 419–420]. For each subproblem, we can apply the eigenvector matrix of the corresponding symmetric arrowhead matrix directly to Y. The cost for each subproblem is $O(Nn)$ using FMM, and there are $O(N/n)$ subproblems at each level. Thus the cost at each level is $O(N^2)$ and the total time is $O(N^2 \log_2 N)$.

When ADC is used to compute only the eigenvalues, the main cost for each subproblem is computing two vectors of the form $q^T U$, finding all the roots of the reformulated secular equation, and computing $\hat{z}$. We now show how to find all the eigenvalues of H and all the components of $\hat{z}$ in $O(n)$ time.

[6] The constant hidden in the $O(\cdot)$ notation depends on the logarithm of the precision.

[7] $\tilde{X}_1$ is also a block-structured matrix (see (18)). Here we view it as a dense matrix to simplify the presentation, even though FMM is more efficient when it exploits this structure.

A root-finder computes successive approximations to each eigenvalue λ_i. The main cost is in evaluating the function[8]

$$f(\lambda) = \lambda - \alpha + \sum_{j=2}^{n} \frac{z_j^2}{d_j - \lambda}.$$

To compute new approximations to all the eigenvalues simultaneously, we must evaluate $f(\lambda)$ at n points. Since this function is similar to the form (20), we can do this in $O(n)$ time using FMM. Thus, assuming that the number of approximations to each eigenvalue is bounded, all the eigenvalues of H can be computed in $O(n)$ time.

To compute $\hat{z}$, note that (7) can be rewritten as

$$|\hat{z}_i| = \sqrt{(d_i - \hat{\lambda}_1)\,(\hat{\lambda}_n - d_i)}\ \exp(\Phi_3(d_i))$$

where

$$\Phi_3(d_i) = \frac{1}{2}\left(\sum_{j=2}^{n-1} \log(\hat{\lambda}_j - d_i) - \sum_{j=2}^{i-1} \log(d_j - d_i) - \sum_{j=i}^{n-1} \log(d_{j+1} - d_i)\right).$$

Thus we can compute all the components of $\hat{z}$ in $O(n)$ time using FMM.

We have shown that when computing all the eigenvalues of T using ADC, we can solve each subproblem in $O(n)$ time. Since there are $O(N/n)$ subproblems at each level, the cost at each level is $O(N)$ and thus the total time is $O(N \log_2 N)$.

6. Numerical results. In this section we compare ADC with three other algorithms for solving the symmetric tridiagonal eigenproblem.

- B/II: Bisection with inverse iteration [21], [23] (subroutines `DSTEBZ` and `DSTEIN` from LAPACK [2]).
- CDC: Cuppen's divide-and-conquer algorithm [11], [12] (subroutine `TREEQL` from `netlib`).
- QR: The QR algorithm [8] (subroutine `DSTEQR` from LAPACK [2]).

ADC solves subproblems of size $N \leq 6$ using the QR algorithm. Since none of the test matrices is particularly large, FMM was not used.

All codes are written in FORTRAN and were compiled with optimization enabled. All computations were done on a SPARCstation/1 in double precision. The machine precision is $\epsilon = 1.1 \times 10^{-16}$.

Let $[\beta, \alpha_i, \beta]$ denote the $N \times N$ symmetric tridiagonal matrix with β on the off-diagonals and $\alpha_1, \ldots, \alpha_N$ on the diagonal. We use the following test matrices, most of which are taken from [21]:

- a random matrix, where the diagonal and off-diagonal elements are uniformly distributed in $[-1, 1]$;
- the Wilkinson matrix $W_N^+ = [1, w_i, 1]$, where $w_i = |(N+1)/2 - i|$;
- a glued Wilkinson matrix W_g^+: a 25×25 block matrix, where each diagonal block is the Wilkinson matrix W_k^+ and the off-diagonal elements $\beta_{i \times k+1} = g$, for $i = 1, \ldots, 24$;

[8] For simplicity we consider the original secular equation. See [17] and [18] for a version of FMM that can compute each $g_i(\mu)$ (and $\psi_i(\mu)$ and $\phi_i(\mu)$ and their derivatives) at a different point in $O(n)$ time. This is needed for the root-finders in [9], [15], [25] and to check the stopping criterion (9).

TABLE 1
Execution time.

Matrix type	Order N	Execution time (seconds) ADC	B/II	CDC	QR
Random	128	3.12	8.50	3.90	11.63
	256	10.43	33.35	14.88	85.86
	512	20.89	133.61	34.31	654.52
W_N^+	129	1.44	6.54	1.46	9.87
	257	3.43	25.00	3.74	66.86
	513	8.26	97.57	14.76	497.55
$W_{10^{-14}}^+$	125	0.63	5.88	*	5.12
	275	2.22	28.83	*	47.35
	525	8.23	121.84	*	353.41
$[1, 2, 1]$	128	3.91	8.49	3.72	10.21
	256	21.89	33.68	22.77	72.40
	512	138.79	144.43	213.01	545.05
$[1, \gamma_i, 1]$	128	4.48	8.54	6.66	10.17
	256	24.20	33.64	43.02	72.14
	512	148.95	135.48	302.06	544.65
$[1/100, 1+\gamma_i, 1/100]$	128	4.57	16.93	6.86	9.83
	256	24.45	102.81	43.01	70.65
	512	149.50	692.64	301.58	539.48

TABLE 2
Residual.

Matrix type	Order N	$\frac{\max_i \|T\hat{x}_i - \hat{\lambda}_i\hat{x}_i\|_2}{N\epsilon\|T\|_2}$ ADC	B/II	CDC	QR
Random	128	0.49×10^{-1}	0.11×10^{-1}	0.10×10^{1}	0.16×10^{0}
	256	0.43×10^{-1}	0.47×10^{-2}	0.74×10^{0}	0.82×10^{-1}
	512	0.23×10^{-1}	0.28×10^{-2}	0.13×10^{1}	0.69×10^{-1}
W_N^+	129	0.67×10^{-1}	0.86×10^{-2}	0.59×10^{0}	0.61×10^{-1}
	257	0.17×10^{-1}	0.39×10^{-2}	0.15×10^{0}	0.35×10^{-1}
	513	0.44×10^{-2}	0.21×10^{-2}	0.67×10^{0}	0.21×10^{-1}
$W_{10^{-14}}^+$	125	0.11×10^{0}	0.16×10^{0}	*	0.22×10^{0}
	275	0.27×10^{-1}	0.36×10^{-1}	*	0.11×10^{0}
	525	0.15×10^{-1}	0.66×10^{-1}	*	0.14×10^{0}
$[1, 2, 1]$	128	0.41×10^{-1}	0.70×10^{-2}	0.31×10^{-1}	0.52×10^{-1}
	256	0.22×10^{-1}	0.12×10^{-1}	0.25×10^{-1}	0.35×10^{-1}
	512	0.12×10^{-1}	0.35×10^{-2}	0.20×10^{-1}	0.25×10^{-1}
$[1, \gamma_i, 1]$	128	0.46×10^{-1}	0.16×10^{-1}	0.67×10^{-1}	0.90×10^{-1}
	256	0.23×10^{-1}	0.11×10^{-1}	0.47×10^{-1}	0.64×10^{-1}
	512	0.12×10^{-1}	0.79×10^{-2}	0.36×10^{-1}	0.47×10^{-1}
$[1/100, 1+\gamma_i, 1/100]$	128	0.22×10^{-1}	0.79×10^{-2}	0.11×10^{-1}	0.60×10^{-1}
	256	0.12×10^{-1}	0.42×10^{-2}	0.11×10^{-1}	0.38×10^{-1}
	512	0.59×10^{-2}	0.21×10^{-2}	0.64×10^{-2}	0.28×10^{-1}

TABLE 3
Orthogonality.

Matrix type	Order N	$\frac{\max_i \|X^T\hat{x}_i - e_i\|_2}{N\epsilon}$ ADC	B/II	CDC	QR
Random	128	0.94×10^{-1}	0.30×10^{0}	0.54×10^{-1}	0.59×10^{0}
	256	0.66×10^{-1}	0.86×10^{-1}	0.17×10^{0}	0.54×10^{0}
	512	0.35×10^{-1}	0.72×10^{-1}	0.30×10^{0}	0.47×10^{0}
W_N^+	129	0.78×10^{-1}	0.35×10^{-1}	0.54×10^{-1}	0.80×10^{0}
	257	0.39×10^{-1}	0.19×10^{-1}	0.89×10^{-1}	0.13×10^{1}
	513	0.19×10^{-1}	0.12×10^{-1}	0.72×10^{-1}	0.13×10^{1}
$W_{10^{-14}}^+$	125	0.64×10^{-1}	0.56×10^{-1}	*	0.38×10^{0}
	275	0.33×10^{-1}	0.16×10^{0}	*	0.31×10^{0}
	525	0.20×10^{-1}	0.34×10^{-1}	*	0.32×10^{0}
$[1, 2, 1]$	128	0.70×10^{-1}	0.78×10^{0}	0.36×10^{0}	0.13×10^{0}
	256	0.47×10^{-1}	0.35×10^{0}	0.18×10^{0}	0.70×10^{-1}
	512	0.39×10^{-1}	0.21×10^{0}	0.14×10^{0}	0.44×10^{-1}
$[1, \gamma_i, 1]$	128	0.62×10^{-1}	0.92×10^{0}	0.17×10^{0}	0.12×10^{0}
	256	0.49×10^{-1}	0.12×10^{1}	0.21×10^{0}	0.62×10^{-1}
	512	0.35×10^{-1}	0.55×10^{0}	0.76×10^{0}	0.48×10^{-1}
$[1/100, 1 + \gamma_i, 1/100]$	128	0.78×10^{-1}	0.35×10^{-1}	0.91×10^{-1}	0.12×10^{0}
	256	0.62×10^{-1}	0.23×10^{-1}	0.11×10^{0}	0.78×10^{-1}
	512	0.61×10^{-1}	0.21×10^{-1}	0.93×10^{-1}	0.40×10^{-1}

- the Toeplitz matrix $[1, 2, 1]$;
- the matrix $[1, \gamma_i, 1]$, where $\gamma_i = i \times 10^{-6}$;
- the matrix $[1/100, 1 + \gamma_i, 1/100]$, where $\gamma_i = i \times 10^{-6}$;
- the test matrices of types 8–21 in the LAPACK test suite.[9]

W_N^+ has pairs of close eigenvalues, W_g^+ has clusters of 50 close eigenvalues, $[1, 2, 1]$ has no close eigenvalues, $[1, \alpha_i, 1]$ and $[1/100, 1 + \alpha_i, 1/100]$ do not deflate, and $[1/100, 1 + \alpha_i, 1/100]$ forces B/II to reorthogonalize all of the eigenvectors.

The numerical results are presented in Tables 1–4. An asterisk means that the algorithm failed. Since the numerical results in Tables 1–3 suggest that CDC and QR are not as competitive, we only compare ADC with B/II for the LAPACK test matrices (see Table 4).

The residual and orthogonality measures for ADC are always comparable with those for QR and B/II, and ADC is roughly twice as fast as CDC for large matrices, due to the differences in how deflation is implemented (see §4.2). In most cases ADC is faster than the others by a considerable margin and in many cases is more than 5–10 times faster. When ADC is slower than B/II (by at most 10%), the matrix size is large ($N \approx 512$) and there are few deflations. These are cases where FMM would make ADC significantly faster.

Acknowledgment. The results in §3 were first announced in a preprint of [20]. Using the ideas there, Borges and Gragg [7] independently derived similar results.

[9] Types 1–7 are all diagonal matrices.

TABLE 4
LAPACK test matrices.

Matrix type	Order N	Execution time		$\frac{\max_i \|T\hat{x}_i - \hat{\lambda}_i\hat{x}_i\|_2}{N\epsilon\|T\|_2}$		$\frac{\max_i \|X^T\hat{x}_i - e_i\|_2}{N\epsilon}$	
		ADC	B/II	ADC	B/II	ADC	B/II
8	128	4.64	8.68	0.47×10^{-1}	0.82×10^{-2}	0.86×10^{-1}	0.16×10^{0}
	256	24.27	33.63	0.27×10^{-1}	0.54×10^{-2}	0.66×10^{-1}	0.25×10^{0}
	512	140.04	133.58	0.17×10^{-1}	0.30×10^{-2}	0.47×10^{-1}	0.21×10^{0}
9	128	2.32	12.66	0.13×10^{0}	0.14×10^{-1}	0.12×10^{0}	0.20×10^{-1}
	256	9.37	76.99	0.28×10^{-1}	0.30×10^{-2}	0.53×10^{-1}	0.27×10^{-1}
	512	44.62	517.45	0.12×10^{-1}	0.92×10^{-1}	0.33×10^{-1}	0.84×10^{-1}
10	128	0.01	12.36	0.11×10^{-1}	0.10×10^{-1}	0.14×10^{-1}	0.70×10^{0}
	256	0.04	84.54	0.45×10^{-2}	0.70×10^{-2}	0.78×10^{-2}	0.52×10^{-1}
	512	0.17	613.86	0.38×10^{-2}	0.21×10^{-2}	0.78×10^{-2}	0.21×10^{-1}
11	128	5.24	8.64	0.45×10^{-1}	0.11×10^{-1}	0.62×10^{-1}	0.22×10^{0}
	256	25.88	33.52	0.28×10^{-1}	0.53×10^{-2}	0.55×10^{-1}	0.17×10^{0}
	512	144.37	132.32	0.17×10^{-1}	0.29×10^{-2}	0.41×10^{-1}	0.20×10^{0}
12	128	4.54	8.75	0.46×10^{-1}	0.85×10^{-2}	0.86×10^{-1}	0.19×10^{0}
	256	24.44	33.94	0.25×10^{-1}	0.54×10^{-2}	0.51×10^{-1}	0.30×10^{0}
	512	141.57	133.83	0.16×10^{-1}	0.31×10^{-2}	0.47×10^{-1}	0.20×10^{0}
13	128	4.61	8.66	0.43×10^{-1}	0.69×10^{-2}	0.62×10^{-1}	0.33×10^{0}
	256	23.49	33.53	0.20×10^{-1}	0.53×10^{-2}	0.70×10^{-1}	0.15×10^{0}
	512	131.57	132.03	0.13×10^{-1}	0.24×10^{-2}	0.41×10^{-1}	0.25×10^{0}
14	128	5.16	8.61	0.46×10^{-1}	0.79×10^{-2}	0.12×10^{0}	0.28×10^{0}
	256	24.88	33.55	0.24×10^{-1}	0.50×10^{-2}	0.51×10^{-1}	0.39×10^{0}
	512	134.86	131.96	0.12×10^{-1}	0.24×10^{-2}	0.43×10^{-1}	0.29×10^{0}
15	128	4.50	8.71	0.49×10^{-1}	0.73×10^{-2}	0.78×10^{-1}	0.12×10^{0}
	256	23.44	33.99	0.24×10^{-1}	0.45×10^{-2}	0.41×10^{-1}	0.17×10^{0}
	512	131.71	133.53	0.13×10^{-1}	0.26×10^{-2}	0.41×10^{-1}	0.13×10^{0}
16	128	4.54	8.68	0.22×10^{-1}	0.87×10^{-2}	0.11×10^{0}	0.11×10^{0}
	256	24.18	33.73	0.13×10^{-1}	0.41×10^{-2}	0.64×10^{-1}	0.93×10^{-1}
	512	139.29	132.47	0.14×10^{-1}	0.19×10^{-2}	0.35×10^{-1}	0.15×10^{0}
17	128	2.67	12.88	0.30×10^{-1}	0.30×10^{-2}	0.11×10^{0}	0.58×10^{-1}
	256	11.78	76.55	0.38×10^{-1}	0.31×10^{-2}	0.51×10^{-1}	0.82×10^{-1}
	512	63.23	521.70	0.16×10^{-1}	0.17×10^{-2}	0.33×10^{-1}	0.29×10^{-1}
18	128	0.01	12.34	0.13×10^{-1}	0.78×10^{-2}	0.16×10^{-1}	0.39×10^{-1}
	256	0.04	83.95	0.98×10^{-2}	0.39×10^{-2}	0.59×10^{-2}	0.29×10^{-1}
	512	0.17	614.26	0.44×10^{-2}	0.19×10^{-2}	0.20×10^{-2}	0.16×10^{0}
19	128	5.08	8.58	0.25×10^{-1}	0.79×10^{-2}	0.90×10^{-1}	0.92×10^{-1}
	256	25.47	33.32	0.14×10^{-1}	0.40×10^{-2}	0.70×10^{-1}	0.16×10^{0}
	512	142.16	131.26	0.13×10^{-1}	0.20×10^{-2}	0.31×10^{-1}	0.97×10^{-1}
20	128	4.46	8.68	0.20×10^{-1}	0.75×10^{-2}	0.62×10^{-1}	0.10×10^{0}
	256	24.12	33.72	0.13×10^{-1}	0.47×10^{-2}	0.51×10^{-1}	0.17×10^{0}
	512	139.29	132.75	0.13×10^{-1}	0.21×10^{-2}	0.33×10^{-1}	0.12×10^{0}
21	128	1.85	12.86	0.45×10^{-1}	0.34×10^{-2}	0.70×10^{-1}	0.45×10^{-1}
	256	6.26	75.81	0.38×10^{-1}	0.27×10^{-2}	0.35×10^{-1}	0.16×10^{-1}
	512	21.07	517.17	0.15×10^{-1}	0.12×10^{-2}	0.31×10^{-1}	0.20×10^{-1}

REFERENCES

[1] L. ADAMS AND P. ARBENZ, *Towards a divide and conquer algorithm for the real nonsymmetric eigenvalue problem*, Tech. Report No. 91-8, Department of Applied Mathematics, University of Washington, Seattle, Aug. 1991.

[2] E. ANDERSON, Z. BAI, C. BISCHOF, J. DEMMEL, J. DONGARRA, J. DU CROZ, A. GREENBAUM, S. HAMMARLING, A. MCKENNEY, S. OSTROUCHOV, AND D. SORENSEN, *LAPACK Users' Guide*, Society for Industrial and Applied Mathematics, Philadelphia, PA, 1992.

[3] P. ARBENZ, *Divide-and-conquer algorithms for the bandsymmetric eigenvalue problem*, Parallel Comput., 18 (1992), pp. 1105–1128.

[4] P. ARBENZ AND G. H. GOLUB, *QR-like algorithms for symmetric arrow matrices*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 655–658.

[5] J. L. BARLOW, *Error analysis of update methods for the symmetric eigenvalue problem*, SIAM J. Matrix Anal. Appl., 14 (1993), pp. 598–618.

[6] D. BOLEY AND G. H. GOLUB, *Inverse eigenvalue problems for band matrices*, in Numerical Analysis, Proceedings, Biennial Conference, Dundee 1977, G. A. Watson, ed., Vol. 630, Lecture Notes in Mathematics, Springer-Verlag, New York, 1977, pp. 23–31.

[7] C. F. BORGES AND W. B. GRAGG, *A parallel divide and conquer algorithm for the generalized real symmetric definite tridiagonal eigenproblem*, in Numerical Linear Algebra and Scientific Computing, L. Reichel, A. Ruttan, and R. S. Varga, eds., de Gruyter, Berlin, 1993, pp. 10–28.

[8] H. BOWDLER, R. S. MARTIN, C. REINSCH, AND J. WILKINSON, *The QR and QL algorithms for symmetric matrices*, Numer. Math., 11 (1968), pp. 293–306.

[9] J. R. BUNCH, C. P. NIELSEN, AND D. C. SORENSEN, *Rank-one modification of the symmetric eigenproblem*, Numer. Math., 31 (1978), pp. 31–48.

[10] J. CARRIER, L. GREENGARD, AND V. ROKHLIN, *A fast adaptive multipole algorithm for particle simulations*, SIAM J. Sci. Statist. Comput., 9 (1988), pp. 669–686.

[11] J. J. M. CUPPEN, *A divide and conquer method for the symmetric tridiagonal eigenproblem*, Numer. Math., 36 (1981), pp. 177–195.

[12] J. J. DONGARRA AND D. C. SORENSEN, *A fully parallel algorithm for the symmetric eigenvalue problem*, SIAM J. Sci. Statist. Comput., 8 (1987), pp. s139–s154.

[13] K. GATES, *Divide and Conquer Methods for the Symmetric Tridiagonal Eigenproblem*, Ph.D. thesis, University of Washington, Seattle, 1991.

[14] G. H. GOLUB AND C. F. VAN LOAN, *Matrix Computations*, The Johns Hopkins University Press, Baltimore, MD, second ed., 1989.

[15] W. B. GRAGG, J. R. THORNTON, AND D. D. WARNER, *Parallel divide and conquer algorithms for the symmetric tridiagonal eigenproblem and bidiagonal singular value problem*, in Modelling and Simulation, W. G. Vogt and M. H. Mickle, eds., Vol. 23, Part 1, University of Pittsburgh School of Engineering, Pittsburgh, 1992, pp. 49–56.

[16] L. GREENGARD AND V. ROKHLIN, *A fast algorithm for particle simulations*, J. Comput. Phys., 73 (1987), pp. 325–348.

[17] M. GU, *Studies in Numerical Linear Algebra*, Ph.D. thesis, Department of Computer Science, Yale University, New Haven, CT, 1993.

[18] M. GU AND S. C. EISENSTAT, *A fast algorithm for updating the singular value decomposition*, manuscript.

[19] ———, *A fast divide-and-conquer method for the symmetric tridiagonal eigenproblem.* Presented at the Fourth SIAM Conference on Applied Linear Algebra, Minneapolis, MN, Sept. 1991.

[20] ———, *A stable and efficient algorithm for the rank-one modification of the symmetric eigenproblem*, SIAM J. Matrix Anal. Appl., 15 (1994), pp. 1266–1276.

[21] E. R. JESSUP, *Parallel Solution of the Symmetric Tridiagonal Eigenproblem*, Ph.D. thesis, Department of Computer Science, Yale University, New Haven, CT, 1989.

[22] ———, *A case against a divide and conquer approach to the nonsymmetric eigenvalue problem*, Applied Numerical Mathematics, 12 (1993), pp. 403–420.

[23] E. R. JESSUP AND I. C. F. IPSEN, *Improving the accuracy of inverse iteration*, SIAM J. Sci. Statist. Comput., 13 (1992), pp. 550–572.

[24] W. KAHAN, *Rank-1 perturbed diagonal's eigensystem*, manuscript, 1989.

[25] R.-C. LI, *Solving secular equations stably and efficiently*, Working paper, Department of Mathematics, University of California at Berkeley, Oct. 1992.

[26] D. P. O'Leary and G. W. Stewart, *Computing the eigenvalues and eigenvectors of symmetric arrowhead matrices*, J. Comput. Phys., 90 (1990), pp. 497–505.
[27] B. N. Parlett and B. Nour-Omid, *The use of a refined error bound when updating eigenvalues of tridiagonals*, Linear Algebra Appl., 68 (1985), pp. 179–219.
[28] W. E. Shreve and M. R. Stabnow, *An eigenvalue algorithm for symmetric bordered diagonal matrices*, in Current Trends in Matrix Theory, F. Uhlig and R. Grone, eds., Elsevier Science Publishing Co., Inc., New York, 1987, pp. 339–346.
[29] D. C. Sorensen and P. T. P. Tang, *On the orthogonality of eigenvectors computed by divide-and-conquer techniques*, SIAM J. Numer. Anal., 28 (1991), pp. 1752–1775.
[30] J. H. Wilkinson, *The Algebraic Eigenvalue Problem*, Clarendon Press, Oxford, 1965.