UCLA
COMPUTATIONAL AND APPLIED MATHEMATICS

A Stable Divide and Conquer Algorithm for the Unitary Eigenproblem

Robert Guzzo
Ming Gu

August 1999

CAM Report 99-25

Department of Mathematics
University of California, Los Angeles
Los Angeles, CA. 90095-1555

http://www.math.ucla.edu/applied/cam/index.html

A STABLE DIVIDE AND CONQUER ALGORITHM FOR THE UNITARY EIGENPROBLEM

ROBERT GUZZO* AND MING GU†

Abstract. We present a Divide and Conquer algorithm for computing the eigendecomposition of a unitary upper Hessenberg matrix H. Previous D & C approaches suffer a potential loss of orthogonality among the computed eigenvectors of H. Using a backward stable method based on previous work by Gu and Eisenstat in the rank-one modification of the symmetric eigenproblem, our algorithm provides a backward stable method for computing the eigenvectors. The method also compares well against the efficiency of other available methods.

1. Introduction.

1.1. Problem Defined.

In this paper, we describe a stable and efficient method for determining the spectral resolution of a unitary[1] upper Hessenberg matrix H of order n:

$$H = \begin{bmatrix} -\bar{\gamma}_0\gamma_1 & -\bar{\gamma}_0\sigma_1\gamma_2 & -\bar{\gamma}_0\sigma_1\sigma_2\gamma_3 & \cdots & -\bar{\gamma}_0\sigma_1\cdots\sigma_{n-1}\gamma_n \\ \sigma_1 & -\bar{\gamma}_1\gamma_2 & -\bar{\gamma}_1\sigma_2\gamma_3 & \cdots & -\bar{\gamma}_1\sigma_2\cdots\sigma_{n-1}\gamma_n \\ & \sigma_2 & -\bar{\gamma}_2\gamma_3 & & \vdots \\ & & \ddots & \ddots & \vdots \\ & & & \sigma_{n-1} & -\bar{\gamma}_{n-1}\gamma_n \end{bmatrix}, \tag{1.1}$$

where σ_k are real and positive, $|\gamma_k|^2 + \sigma_k^2 = 1$ for $1 \le k < n, \gamma_0 = 1$ and $|\gamma_n| = 1$ [16]. We call the γ_k's the *Schur parameters* of H, and the σ_k's the *complementary parameters* of H.

We seek the spectral resolution of H:

$$H = W\Omega W^*, \tag{1.2}$$

where the columns of the matrix W are the eigenvectors of H; and Ω is a diagonal matrix whose diagonal entries are the eigenvalues corresponding to the eigenvectors in W. It is easy to show that since H is unitary, W can also be chosen to be unitary and the eigenvalues of H must have unit modulus [11].

There are two general methods available for calculating the spectral resolution of H: QR algorithms and Divide & Conquer algorithms. Various QR algorithms have been developed which compute solutions to the eigenproblem in a stable fashion. Recent work by Ammar [4], Gragg [9, 12], and Stewart [16] has shown certain QR algorithms to be quite stable. However, there are certain advantages to Divide & Conquer strategies proposed by Ammar, Reichel and Sorensen [2], Gragg and Reichel [11]. Namely, such methods can be implemented much more efficiently and are better-suited to parallel implementation. In fact, such strategies have been used to solve the symmetric tridiagonal eigenvalue problem with great success (see Cuppen [6], Dongarra and Sorensen [7], Gu and Eisenstat [13, 14]).

*Department of Mathematics, University of California, Los Angeles. Email address: rguzzo@math.ucla.edu. This research was supported in part by NSF Career Award CCR-9702866.

†Department of Mathematics, University of California, Los Angeles. Email address: mgu@math.ucla.edu. This research was supported in part by NSF Career Award CCR-9702866 and by Alfred Sloan Research Fellowship BR-3720.

[1] A unitary matrix H satisfies that $H^*H = HH^* = I$, where $*$ denotes complex conjugate transposition.

The traditional problem with D & C methods is numerical instability, especially in regards to calculating the eigenvectors of H (see Ammar, Reichel and Sorensen [2] and Stewart [16]). On the contrary, the method presented here will be numerically stable, guaranteeing that the columns of W are numerically orthogonal and that the eigenvalues of H all lie on the unit circle in the complex plane. Our extensive numerical experiments indicate that our method compares very well against existing methods in both efficiency and accuracy (see § 5).

It is helpful to note that the interest surrounding this problem arises out of signal processing applications [3], more specifically, in frequency estimation, including Pisarenko's method [1]. The applications to signal processing are closely related to the computation of Gauss-Szegö quadrature rules, which is discussed more fully in [11].

Throughout the paper, we use the usual model of floating point arithmetic:

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

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

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

for any positive floating point number x.

Let $\hat{x}$ be an approximation to $x \neq 0$. For the purpose of this paper, we say that $\hat{x}$ is close to x (to high absolute accuracy) if $x - \hat{x} = O(\epsilon)$; and we say that $\hat{x}$ is close to $x \neq 0$ to high relative accuracy if $(x - \hat{x})/x = O(\epsilon)$. Finally, we shall let $\|\cdot\|$ denote the vector 2-norm.

The rest of the paper is organized as follows. In § 2, we introduce the Unitary Divide and Conquer (UDC) algorithm presented in [2, 10], which is referred to as "old UDC" or "original UDC" in this paper. This algorithm is a FORTRAN implementation of the method introduced by Gragg and Reichel [10, 11]. In the same section, we will also introduce our new method, referred to as "new UDC" or "our UDC algorithm." The new UDC is a modification of the old UDC, extending previous work by Gu and Eisenstat in [13, 14]. In § 3, we discuss the nature of the rootfinder used in the new method as well as providing a specific way to handle eigenvalues. In § 4 we prove the numerical stability of our method. Finally, in § 5, we will present some numerical results for various types of eigenproblems.

2. Solving the Unitary Eigenproblem Recursively. From the Schur parameters and complementary parameters of H in (1.1), we can uniquely represent H in its *Schur parametric form* [16]:

$$H = H(\gamma_1, \gamma_2, \ldots, \gamma_n) = G_1 G_2 \cdots G_{n-1} \widetilde{G}_n, \tag{2.1}$$

where each $G_k \in \mathbf{C}^{\mathbf{n}\times\mathbf{n}}, 1 \leq k < n$, is a Givens matrix,

$$G_k = \begin{bmatrix} I_{k-1} & & & \\ & -\gamma_k & \sigma_k & \\ & \sigma_k & \bar{\gamma}_k & \\ & & & I_{n-k-1} \end{bmatrix}, \quad \gamma_k \in \mathbf{C},\ \sigma_k \in \mathbf{R},\ \sigma_k \geq 0,\ |\gamma_k|^2 + \sigma_k^2 = 1,$$

and $\widetilde{G}_n$ is the diagonal matrix

$$\widetilde{G}_n = \begin{bmatrix} I_{n-1} & \\ & -\gamma_n \end{bmatrix}, \quad \gamma_n \in \mathbf{C}, |\gamma_n| = 1.$$

Given the matrix H in upper Hessenberg form, it is easy to compute the Schur parameters (for details, see [11]). Working with the Schur parameters and complementary parameters of H, instead of with H itself will greatly reduce the computational complexity of the algorithm. It would appear that we can further reduce the amount of storage necessary by only storing the γ_k values, and calculating the σ_k values as needed. However, this calculation could lead to numerical instability should any of the $|\gamma_k|$ be close to one (see Stewart [16]).

2.1. The Divide Phase. The idea behind Divide and Conquer is to obtain the spectral resolution of H from the spectral resolution of two subproblems. As described in [2] (details in [11]), we will make use of the fact that a complex Givens matrix G_s is diagonally unitarily equivalent with a real Givens reflector, which can be written as a Householder transformation. Define

$$\gamma_s^{'} = \begin{cases} \gamma_s/|\gamma_s|, & \gamma_s \neq 0, \\ 1, & \gamma_s = 0. \end{cases}$$

Then $|\gamma_s^{'}| = 1$ and

$$\begin{bmatrix} I_{s-1} & & \\ & \bar{\gamma}_s^{'} & \\ & & I_{n-s} \end{bmatrix} G_s \begin{bmatrix} I_s & & \\ & \gamma_s^{'} & \\ & & I_{n-s-1} \end{bmatrix} = \begin{bmatrix} I_{s-1} & & & \\ & -|\gamma_s| & \sigma_s & \\ & \sigma_s & |\gamma_s| & \\ & & & I_{n-s-1} \end{bmatrix}.$$

The right hand side above can be written as a Householder transformation $I - 2ww^*$, where $w \in \mathbf{R}^n$ satisfies

$$w = \omega_s e_s + \omega_{s+1} e_{s+1} \text{ with } \omega_s = ((1+|\gamma_s|)/2)^{1/2}, \;\; \omega_{s+1} = -\sigma_s/(2(1+|\gamma_s|))^{1/2},$$

and e_j denotes the j^{th} axis vector of length n. We can now express (2.1) as two subproblems "pasted" together by the Householder transformation:

$$H = \begin{bmatrix} H_1 & \\ & I_{n-s} \end{bmatrix} (I - 2ww^*) \begin{bmatrix} I_s & \\ & H_2 \end{bmatrix}, \tag{2.2}$$

where, using the notation in (2.1),

$$\begin{aligned} H_1 &= H(\gamma_1, \ldots, \gamma_{s-1}, -\gamma_s^{'}) \in \mathbf{C}^{s \times s}, \\ H_2 &= H(\bar{\gamma}_s^{'}\gamma_{s+1}, \bar{\gamma}_s^{'}\gamma_{s+2}, \ldots, \bar{\gamma}_s^{'}\gamma_n) \in \mathbf{C}^{(n-s)\times(n-s)}. \end{aligned}$$

For the purposes of Divide and Conquer, we assume that we know the spectral resolutions of the two submatrices:

$$H_k = W_k \Lambda_k W_k^*, \qquad k = 1, 2.$$

where the W_k are unitary and the Λ_k are diagonal. Now we seek the spectral resolution of the original matrix, H. Define

$$\widetilde{W} = \begin{bmatrix} W_1 & \\ & W_2 \end{bmatrix}, \;\; \Lambda = diag(\lambda_1, \lambda_2, \ldots, \lambda_n) = \begin{bmatrix} \Lambda_1 & \\ & \Lambda_2 \end{bmatrix}, \;\; z = \begin{bmatrix} W_1^* e_s \omega_s \\ \bar{\Lambda}_2 W_2^* e_{s+1} \omega_{s+1} \end{bmatrix}.$$

Note that $z^* z = 1$, in exact arithmetic. Substitution of above into (2.2) renders the following formulation:

$$H = \widetilde{W} \Lambda (I - 2zz^*) \widetilde{W}^*. \tag{2.3}$$

Since $\widetilde{W}$ is unitary, (2.3) reveals that H and the core matrix

$$A = \Lambda(I - 2zz^*) \quad \text{with} \quad z^*z = 1 \tag{2.4}$$

have the same eigenvalues. Let $A = U\Lambda' U^*$ be the eigendecomposition of A. Then the eigendecomposition (1.2) for H is simply

$$\Omega = \Lambda' \quad \text{and} \quad W = \widetilde{W}U. \tag{2.5}$$

Since A is a rank-one modification on diagonal Λ, we will determine the spectral resolution of A by using a similar strategy as in [13].

To complete the divide phase, we choose, say, $s = \lfloor n/2 \rfloor$. We then recursively apply the dividing strategy (2.2) to H_1 and H_2, respectively, until their dimensions are sufficiently small (less than 10, for example), resulting in $O(\log_2 n)$ levels of recursion. We can obtain the spectral resolution of the sufficiently small problems at the bottom of the recursion tree directly with little effort. To obtain the spectral resolution of the original matrix H in (1.1), we solve the core problems of the form (2.4) at every level of the recursion tree in a bottom-up fashion. The eigenvectors of H can be recursively computed as $\widetilde{W}U$ (see (2.5)). The total cost for this algorithm is $O(n^3)$ flops[2]. Note that the actual cost of this algorithm can sometimes be much lower due to deflation (see § 2.3).

Similar to the Divide and Conquer methods for the symmetric tridiagonal eigenvalue problem, the above recursion can also be simplified into a Divide and Conquer method for computing the eigenvalues of H only, with a total cost of $O(n^2)$ flops (see Ammar, Reichel and Sorensen [2], Cuppen [6], Dongarra and and Sorensen [7], Gragg and Reichel [11], Gu and Eisenstat [13, 14]).

2.2. Roots of the Spectral Function. To determine the eigenvalues of A, we must find the roots of its characteristic polynomial,

$$\chi(\lambda) = \det(A - \lambda I) = \det(\Lambda - \lambda I)\left(1 + 2z^*(\Lambda - \lambda I)^{-1}\Lambda z\right) = 0.$$

Thus, the eigenvalues of A include the roots of the spectral function,

$$\phi(\lambda) = 1 + 2z^*(\Lambda - \lambda I)^{-1}\Lambda z = 0.$$

Since $z^*z = 1$ according to (2.4), we can rewrite the spectral function as

$$\phi(\lambda) = \sum_{j=1}^{n} |z_j|^2 \frac{\lambda + \lambda_j}{\lambda - \lambda_j} = 0. \tag{2.6}$$

Recall that the eigenvalues for a unitary upper Hessenberg matrix all lie on the unit circle. Hence, the eigenvalues of A and Λ can be written as $\lambda = \exp(i\theta)$ and $\lambda_j = \exp(i\theta_j)$, respectively, where $i = \sqrt{-1}$ and we restrict $-\pi < \theta_j \leq \pi$. Substitution into (2.6) renders

$$\Phi(\theta) = -i\phi(\lambda) = \sum_{j=1}^{n} |z_j|^2 \cot\left(\frac{\theta - \theta_j}{2}\right) = 0. \tag{2.7}$$

[2] A *flop* is a floating point operation $x \circ y$, where x and y are floating point numbers and $\circ$ is one of $+, -, \times$, and $\div$.

Thus, finding the roots of the spectral function is equivalent to finding the roots of $\Phi(\theta)$. Inspection of this function shows that Φ has n poles on the interval $(-\pi, \pi]$, occurring at each of the θ_j's. Also, Φ is a monotonically decreasing function on any interval between two adjacent poles. Following Golub [8], we call equation (2.7) the *secular equation.* We will talk about how to find roots of this equation in § 3.2.

2.3. Deflation. The work of Divide and Conquer methods can be reduced (sometimes dramatically) by the deflation procedure described in [2]. If two diagonal entries of Λ, λ_j and λ_k, are identical or have very close arguments, then λ_j can be regarded as an eigenvalue of A. If some component z_j of z is zero or has very small magnitude, then λ_j can again be regarded as an eigenvalue of A. In both cases, A can be reduced to a matrix with similar structure but smaller dimension. This will reduce the amount of computation involved in finding the eigenvalues, since there are fewer roots of Φ. Used to full advantage, deflation can also reduce the amount of calculation involved in computing the eigenvectors of A and H. More fundamentally, the stability of our method relies on the assumption that deflation has been done (see § 4). Similar deflation procedures have also been used in the numerical solution of the symmetric tridiagonal eigenproblem (see Cuppen [6], Dongarra and Sorensen [7], Gu and Eisenstat [13, 14]).

From now on, we will assume that the deflation procedure of [2] has been applied to A in (2.4). We assume that $n > 1$ and that the θ's are ordered in the following way:

$$-\pi < \theta_1 < \theta_2 < \cdots < \theta_n \leq \pi. \tag{2.8}$$

Our implementation of the deflation procedure ensures that

$$|z_j| \geq \epsilon^{'}, \;\; \theta_{k+1} - \theta_k \geq \epsilon^{''} \;\text{ and }\; (\pi + \theta_1) + (\pi - \theta_n) \geq \epsilon^{''}. \tag{2.9}$$

where $\epsilon^{'}$ and $\epsilon^{''}$ are some specified deflation tolerances, to be discussed in more detail in § 4. The last condition in (2.9) ensures that angles between any two eigenvalues, both in the clockwise direction and counter-clockwise direction around the circle, are at least as big as $\epsilon^{''}$.

The conditions in (2.9) imply that the eigenvalues of H, $\lambda_j^{'}$ for $1 \leq j \leq n$, strictly interlace on the unit circle with the λ_j [2]. Let $\hat{\lambda}_j$ be the computed eigenvalues of H. Since all eigenvalues of H are on the unit circle, we can further write

$$\lambda_j^{'} = \exp\left(i\theta_j^{'}\right) \quad\text{and}\quad \hat{\lambda}_j = \exp\left(i\hat{\theta}_j\right). \tag{2.10}$$

2.4. Unstable Eigenvector Formulas. It follows from (2.4) that the normalized eigenvector of A associated with $\lambda_j^{'}$ satisfies the following formulas:

$$v_j = \frac{u}{\|u\|}, \qquad u = (I - \Lambda^* \lambda_j^{'})^{-1} z. \tag{2.11}$$

The UDC algorithm in [2, 10] computes approximations $\hat{\theta}_j$ by solving (2.7), and computes the eigenvectors of A and H using equation (2.11), with $\lambda_j^{'}$ replaced by $\hat{\lambda}_j$. However, due to the potential ill-conditioning in the eigenvectors, the vectors computed this way can often lose mutual orthogonality in finite precision, leading to inaccurate spectral resolution of A and H (see Ammar, Reichel and Sorensen [2], Gragg and Reichel [11], and Sun [17]). Similar instability problems also occurred in the old Divide and Conquer methods for the symmetric tridiagonal eigenproblem (see Dongarra and and Sorensen [7], Gu and Eisenstat [13, 14]).

2.5. Stable Eigenvector Formulas. To develop a stable method for computing the eigenvectors, we first re-write the k^{th} component of u as follows:

$$\begin{aligned} u_k &= \frac{z_k}{1-\lambda_j'\bar{\lambda}_k} = \frac{z_k}{1-\lambda_j'/\lambda_k} \\ &= \frac{z_k}{1-\cos(\theta_j'-\theta_k)-i\sin(\theta_j'-\theta_k)} \end{aligned}$$

Making use of the double-angle formulas, we find that

$$\begin{aligned} u_k &= \frac{z_k}{2\sin^2\left(\frac{\theta_j'-\theta_k}{2}\right)-2i\sin\left(\frac{\theta_j'-\theta_k}{2}\right)\cos\left(\frac{\theta_j'-\theta_k}{2}\right)} \\ &= \frac{i}{2}\exp\left(i\left(\frac{\theta_k-\theta_j'}{2}\right)\right)\cdot\left(z_k\Big/\sin\left(\frac{\theta_j'-\theta_k}{2}\right)\right) \end{aligned} \tag{2.12}$$

From the above formulation, we observe that the eigenvectors of H can be directly calculated in terms of the poles and roots of the spectral function and the components of z. Furthermore, if θ_j' were known exactly and could be exactly represented as a floating point number, then we would be able to compute u_k to full accuracy using (2.12).

Of course, the angles $\hat{\theta}_j$ computed by our rootfinder by solving (2.7) are only *approximations* to θ_j'. If $\hat{\theta}_j$ is used in place of θ_j' in equation (2.12) to compute u_k, the computed u_k can incur a very large relative error, which can lead to loss of orthogonality among computed eigenvectors. In other words, equation (2.12) is still an unstable way to compute the eigenvectors.

It turns out that a stable method for computing the eigenvectors can be developed by constructing a new matrix

$$\widehat{A} = \Lambda(I-2\gamma\hat{z}\hat{z}^*), \tag{2.13}$$

where γ is a complex scalar. It is clear that $\widehat{A}$ has a similar structure as A. We choose the scalar γ and vector $\hat{z}$ so that the exact eigenvalues of $\widehat{A}$ are the eigenvalues we computed for A. In § 2.6 we will show that this matrix does exist and is in fact unitary with distinct eigenvalues. Hence the eigenvectors of $\widehat{A}$ are always mutually orthogonal.

Similar to formulas (2.11) and (2.12), the eigenvector of $\widehat{A}$ associated with $\hat{\lambda}_j$ can be computed as

$$\hat{v}_j = \frac{\hat{u}}{\|\hat{u}\|}, \tag{2.14}$$

where the k^{th} component of $\hat{u}$ satisfies

$$\hat{u}_k = \frac{i}{2}\exp\left(i\left(\frac{\theta_k-\hat{\theta}_j}{2}\right)\right)\cdot\left(\hat{z}_k\Big/\sin\left(\frac{\hat{\theta}_j-\theta_k}{2}\right)\right). \tag{2.15}$$

Note that γ does not appear in (2.14) and (2.15). It also follows from (2.15) that

$$\|\hat{u}\| = \frac{1}{2}\sqrt{\sum_{k=1}^{n}\left(\hat{z}_k\Big/\sin\left(\frac{\hat{\theta}_j-\theta_k}{2}\right)\right)^2}. \tag{2.16}$$

In § 2.6, we show that the vector $\hat{z}$ can be computed to high relative accuracy from the $\hat{\theta}$'s; and in § 3.2, we show that the denominators in formulas (2.15) and (2.16), $\sin\left(\dfrac{\hat{\theta}_j - \theta_k}{2}\right)$, can also be computed to high relative accuracy. Consequently, we can compute $\|\hat{u}\|$ to high relative accuracy as well. In addition, it is clear that we can compute the unit modulus term $\exp\left(i\left(\dfrac{\theta_k - \hat{\theta}_j}{2}\right)\right)$ in (2.15) to high relative accuracy. It now follows from (2.14) and (2.15) that we can compute the eigenvector $\hat{v}_j$ to high relative accuracy.

The above analysis implies that we can compute all the the eigenvectors of $\widehat{A}$ to high relative accuracy regardless of its eigenvalue distribution. Since $\widehat{A}$ is itself unitary, these computed eigenvectors will be numerically orthogonal. In our UDC algorithm, we use these vectors as approximations to the eigenvectors of A. In § 4, we justify this approach by showing that the matrix $\widehat{A}$ is very close to A in finite precision, hence the spectral resolution for $\widehat{A}$ is a good approximation of that of A. A similar approach has been taken by Gu and Eisenstat in the rank-one modification of the symmetric eigenproblem [13].

2.6. Building $\widehat{A}$. In the following we construct the matrix $\widehat{A}$ by deriving formulas for γ and the components of the vector $\hat{z}$. We assume that the deflation procedure has been performed on A and thus the deflation criteria (2.9) hold.

It has already been stated that the n roots of Φ strictly interlace its poles on the unit circle. Our rootfinder discussed in § 3.2 guarantees that the computed angles $\hat{\theta}$'s satisfy

$$-\pi < \theta_1 < \hat{\theta}_1 < \theta_2 < \hat{\theta}_2 < \cdots < \hat{\theta}_{n-1} < \theta_n < \hat{\theta}_n < \theta_1 + 2\pi. \tag{2.17}$$

Note that the unique $\hat{\theta}_n$ that satisfies (2.17) may actually be greater than π. We will have further discussion on $\hat{\theta}_n$ in § 3.2 (see (3.7)).

The characteristic polynomial for $\widehat{A}$ can be written as follows:

$$\begin{aligned}\widehat{\chi}(\lambda) &= \det(\widehat{A} - \lambda I) = \det(\Lambda - \lambda I)\left(1 - 2\gamma\hat{z}^*(\Lambda - \lambda I)^{-1}\Lambda\hat{z}\right)\\ &= \det(\Lambda - \lambda I)\left(1 - 2\gamma\sum_{j=1}^{n}\frac{\lambda_j}{\lambda_j - \lambda}|\hat{z}_j|^2\right)\\ &= \prod_{j=1}^{n}(\lambda_j - \lambda) - 2\gamma\sum_{j=1}^{n}\left(\frac{\lambda_j|\hat{z}_j|^2\prod_{l=1}^{n}(\lambda_l - \lambda)}{\lambda_j - \lambda}\right)\end{aligned} \tag{2.18}$$

On the other hand, the fact that the $\hat{\lambda}_j$'s are the eigenvalues of $\widehat{A}$ implies that

$$\widehat{\chi}(\lambda) = \prod_{j=1}^{n}(\hat{\lambda}_j - \lambda). \tag{2.19}$$

Combining (2.18) and (2.19), and setting $\lambda = \lambda_k$, for $k = 1, 2, \ldots, n$, we obtain

$$\prod_{j=1}^{n}(\hat{\lambda}_j - \lambda_k) = -2\gamma\lambda_k|\hat{z}_k|^2\prod_{j=1,\ j\neq k}^{n}(\lambda_j - \lambda_k) \tag{2.20}$$

Solving for $\gamma|\hat{z}_k|^2$, and using the same calculations as in (2.12), we get

$$
\begin{aligned}
\gamma|\hat{z}_k|^2 = -\frac{\prod_{j=1}^n(\hat{\lambda}_j-\lambda_k)}{2\lambda_k\prod_{j\neq k}(\lambda_j-\lambda_k)} &= -\frac{\prod_{j=1}^n(\hat{\lambda}_j/\lambda_k-1)}{2\prod_{j\neq k}(\lambda_j/\lambda_k-1)} \\
&= -i\exp\left(i\sum_{j=1}^n\frac{\hat{\theta}_j-\theta_j}{2}\right)\frac{\prod_{j=1}^n\sin((\hat{\theta}_j-\theta_k)/2)}{\prod_{j\neq k}\sin((\theta_j-\theta_k)/2)}. \qquad (2.21)
\end{aligned}
$$

In the following, we discuss how to choose γ and $\hat{z}_k$ according to (2.21). To this end, we rewrite the ratio of products in (2.21) as

$$
\frac{\prod_{j=1}^n\sin((\hat{\theta}_j-\theta_k)/2)}{\prod_{j\neq k}\sin((\theta_j-\theta_k)/2)} = \sin((\hat{\theta}_k-\theta_k)/2)\cdot\left(\prod_{j\neq k}\frac{\sin((\hat{\theta}_j-\theta_k)/2)}{\sin((\theta_j-\theta_k)/2)}\right). \qquad (2.22)
$$

The interlacing property (2.17) implies that

$$
\begin{array}{rcccl}
0 & < & \frac{\hat{\theta}_j-\theta_k}{2},\ \frac{\theta_j-\theta_k}{2} & < & \pi, \quad \text{if } j>k, \\
-\pi & < & \frac{\hat{\theta}_j-\theta_k}{2},\ \frac{\theta_j-\theta_k}{2} & < & 0, \quad \text{if } j<k,
\end{array} \qquad (2.23)
$$

and

$$
0<(\hat{\theta}_k-\theta_k)/2<\pi. \qquad (2.24)
$$

It follows from these relations that the first term in (2.22) must be positive; it also follows that every ratio in the product in (2.22) must be positive. Hence the left hand side in (2.22) must be positive for every k.

This discussion suggests that the following choice of γ and $\hat{z}_k$ satisfies equation (2.21):

$$
|\hat{z}_k| = \sqrt{\frac{\prod_{j=1}^n\sin((\hat{\theta}_j-\theta_k)/2)}{\prod_{j\neq k}\sin((\theta_j-\theta_k)/2)}} \quad \text{and} \quad \gamma = -i\exp\left(i\sum_{j=1}^n\frac{\hat{\theta}_j-\theta_j}{2}\right) \qquad (2.25)
$$

Furthermore, since z_k is usually a complex number, we choose the phase angle of $\hat{z}_k$ to be that of z_k. Hence

$$
\hat{z}_k = |\hat{z}_k|\frac{z_k}{|z_k|}, \quad \text{for} \quad 1\le k\le n. \qquad (2.26)
$$

To complete the construction for $\widehat{A}$, we note that by working the above steps backwards, it is straightforward to verify that equations (2.13), (2.26) and (2.25) indeed uniquely define a matrix $\widehat{A}$ that has the $\hat{\lambda}_j$'s as its eigenvalues.

In the following, we show that $\widehat{A}$ is unitary. According to (2.10),

$$
\begin{aligned}
1 = \left|\prod_{j=1}^n\hat{\lambda}_j\right| &= \left|\det(\widehat{A})\right| = \left|\det\left(\Lambda(I-2\gamma\hat{z}\hat{z}^*)\right)\right| \\
&= |\det(\Lambda)|\cdot|\det(I-2\gamma\hat{z}\hat{z}^*)| = |1-2\gamma\hat{z}^*\hat{z}|\,.
\end{aligned}
$$

The last equation implies that

$$\gamma + \bar{\gamma} - 2|\gamma|^2 \hat{z}^* \hat{z} = 0.$$

Consequently,

$$\begin{aligned} \widehat{A}^* \widehat{A} &= (I - 2\gamma \hat{z}\hat{z}^*)^* \Lambda^* \Lambda (I - 2\gamma \hat{z}\hat{z}^*) \\ &= (I - 2\gamma \hat{z}\hat{z}^*)^* (I - 2\gamma \hat{z}\hat{z}^*) = I - 2\left(\gamma + \bar{\gamma} - 2|\gamma|^2 \hat{z}^* \hat{z}\right) \hat{z}\hat{z}^* = I. \end{aligned}$$

Finally, we note that the components of the vector z can also be expressed in terms of Λ and the eigenvalues of A. Indeed, equation (2.21) now becomes

$$|z_k|^2 = -i \exp\left(i \sum_{j=1}^{n} \frac{\theta_j' - \theta_j}{2} \right) \frac{\prod_{j=1}^{n} \sin((\theta_j' - \theta_k)/2)}{\prod_{j \neq k} \sin((\theta_j - \theta_k)/2)}. \tag{2.27}$$

Since the λ_j' are the eigenvalues of H and since $z^* z = 1$,

$$\begin{aligned} \exp\left(i \sum_{j=1}^{n} \theta_j' \right) &= \prod_{j=1}^{n} \lambda_j' = \det(H) = \det\left(\Lambda(I - 2zz^*)\right) \\ &= -\det(\Lambda) = -\prod_{j=1}^{n} \lambda_j = -\exp\left(i \sum_{j=1}^{n} \theta_j \right). \end{aligned}$$

It follows that

$$\exp\left(i \sum_{j=1}^{n} \frac{\theta_j' - \theta_j}{2} \right) = i. \tag{2.28}$$

In light of above discussion, we can rewrite (2.27) as

$$|z_k| = \sqrt{\frac{\prod_{j=1}^{n} \sin((\theta_j' - \theta_k)/2)}{\prod_{j \neq k} \sin((\theta_j - \theta_k)/2)}}. \tag{2.29}$$

3. Some Computational Issues.

3.1. The FORTRAN Sine Function. Formulas (2.15), (2.16) and (2.25) all involve the sine function. To guarantee numerical stability, we would like to compute every sine term as accurately as we can. Throughout this paper, we assume that

ASSUMPTION 3.1. *The FORTRAN sine function computes* $\sin(\psi)$ *to high relative accuracy for* $|\psi| \leq \pi/2$.

It is not realistic to require the FORTRAN sine function to compute $\sin(\psi)$ to high relative accuracy for any ψ. In fact, since $\sin(\pm\pi) = 0$, large relative errors are hard to avoid for any FORTRAN sine function if ψ is very close to $\pm\pi$.

In the following, we show that for $|\psi| \leq \pi/2$, a small relative change in ψ can only imply a small relative change in $\sin(\psi)$. In fact, for any $|\varepsilon| \ll 1$,

$$\begin{aligned} \sin(\psi(1+\varepsilon)) - \sin\psi &= \sin\psi \cdot (\cos(\psi\varepsilon) - 1) + \sin(\psi\varepsilon) \cdot \cos\psi \\ &= -2 \sin\psi \cdot \sin^2(\psi\varepsilon/2) + \sin(\psi\varepsilon) \cdot \cos\psi. \end{aligned}$$

Taking absolute value,

$$\begin{aligned}|\sin(\psi(1+\varepsilon)) - \sin\psi| &\leq 2\left|\sin\psi \cdot \sin^2(\psi\varepsilon/2)\right| + |\sin(\psi\varepsilon)\cdot\cos\psi| \\ &\leq 2\left|\sin(\psi\varepsilon/2)\right| + |\sin(\psi\varepsilon)| \leq 2\cdot|\psi\varepsilon/2| + |\psi\varepsilon| \\ &= 2\,|\psi\varepsilon| \leq \pi\,|\varepsilon\sin\psi|\,,\end{aligned}$$

where we have used the fact that

$$\frac{2}{\pi} \leq \frac{\sin\psi}{\psi} \leq 1 \quad \text{for} \quad |\psi| \leq \pi/2. \tag{3.1}$$

It now follows that

$$\frac{|\sin(\psi(1+\varepsilon)) - \sin\psi|}{|\sin\psi|} \leq \pi|\varepsilon|. \tag{3.2}$$

Assumption 3.1 and relation (3.2) imply that the sine terms in formulas (2.15), (2.16) and (2.25) can be computed to high relative accuracy if their arguments are between $-\pi/2$ and $\pi/2$ and are computed to high relative accuracy. In § 3.2, we will further discuss how to compute the sine terms in these formulas accurately when the arguments are not between $-\pi/2$ and $\pi/2$.

3.2. The Rootfinder and Computing Angles. Our rootfinder for finding the roots of equation (2.7) is basically the cubically convergent rootfinder developed in [2, 11], with a number of modifications aimed at improving numerical accuracy. We assume that the deflation procedure has been performed on A and thus relations (2.9) and (2.8) hold. The n roots θ_j' of Φ satisfy strict interlacing properties similar to (2.17).

In each interval, (θ_j, θ_{j+1}) for $j < n$, denote

$$\alpha_j = \theta_j' - \theta_j \quad \text{and} \quad \beta_j = \theta_{j+1} - \theta_j'. \tag{3.3}$$

If θ_j' is closer to θ_j, the rootfinder computes an approximation $\hat{\alpha}_j$ to α_j. It then computes $\hat{\beta}_j$, the approximation to β_j, according to the following:

$$\hat{\beta}_j = (\theta_{j+1} - \theta_j) - \hat{\alpha}_j. \tag{3.4}$$

If θ_j' is closer to θ_{j+1}, then the rootfinder computes an approximation $\hat{\beta}_j$ to β_j; it then computes the approximation $\hat{\alpha}_j$ from (3.4).. We will postpone discussion on the computation of θ_n' to the end of § 3.2.

By computing the smaller of the two angles between the root θ_j' and its two nearest poles, we prevent any catastrophic cancellation when the root is extremely close to one of the poles. With $\hat{\alpha}_j$ and $\hat{\beta}_j$, the difference between θ_j' and any pole θ_k can be approximated as

$$\hat{\theta}_j - \theta_k = \begin{cases} \hat{\alpha}_j + (\theta_j - \theta_k), & \text{for } \theta_k \leq \theta_j \\ (\theta_{j+1} - \theta_k) - \hat{\beta}_j, & \text{for } \theta_k > \theta_j \end{cases}$$

This way, we can compute $\hat{\theta}_j - \theta_k$ to high relative accuracy, given $\hat{\alpha}_j$ and $\hat{\beta}_j$. In particular, we avoid any catastrophic cancellation in the event that θ_j' is very close to one of the poles. According to (3.1), we can also compute $\sin((\hat{\theta}_j - \theta_k)/2)$ to high relative accuracy if $|\hat{\theta}_j - \theta_k|/2 \leq \pi/2$. A similar result holds for $\sin((\theta_j - \theta_k)/2)$.

To accurately compute the sine terms in (2.15), (2.16) and (2.25) when the arguments are not between $-\pi/2$ and $\pi/2$, we recall that the eigenvalues all lie on the unit circle. Therefore, calculating angles between eigenvalues can be done in either the clockwise or counter-clockwise direction around the circle. If the angle between two points on the unit circle is calculated in the counter-clockwise direction to be close to 2π, then in the clockwise direction, the angle is close to zero. We achieve this effect when we make the following alternate formulation:

$$\sin(\frac{\theta_j - \theta_k}{2}) = \begin{cases} -\sin\left(\dfrac{\nu_1 + (\theta_j - \theta_1) + (\theta_n - \theta_k) + \nu_n}{2}\right) & \text{if } (\theta_j - \theta_k)/2 < -\pi/2, \\ \sin\left(\dfrac{\nu_n + (\theta_n - \theta_j) + (\theta_k - \theta_1) + \nu_1}{2}\right) & \text{if } (\theta_j - \theta_k)/2 > \pi/2, \end{cases}$$

(3.5)
where $\nu_1 = \pi + \theta_1$ and $\nu_n = \pi - \theta_n$. Given ν_1 and ν_n, the arguments on the right hand side of (3.5) can be computed to high relative accuracy, so can the sine function. We also make a similar reformulation to $\sin((\hat{\theta}_j - \theta_k)/2)$. However, since π is not a floating point number, it sometimes may not be possible to compute ν_1 and ν_n to high relative accuracy. See § 4 for further discussion on their computation.

Now, we address the issue of computing $\theta^{'}_n$. In the spirit of the above discussion, let α_n and β_n be the smaller angles on the circle between $\theta^{'}_n$ and its nearest poles, θ_n and $\theta_1 + 2\pi$. If $\alpha_n \le \beta_n$, then our rootfinder computes an approximation $\hat{\alpha}_n$ by solving (2.7) and computes $\hat{\beta}_n$ from $\hat{\alpha}_n$ using the following formula:

$$\hat{\beta}_n = (\theta_1 + 2\pi - \theta_n) - \hat{\alpha}_n = (\nu_1 + \nu_n) - \hat{\alpha}_n. \tag{3.6}$$

Otherwise, it computes $\hat{\beta}_n$ by solving (2.7) and computes $\hat{\alpha}_n$ from $\hat{\beta}_n$. After $\hat{z}$ and the eigenvectors for $\widehat{A}$ are computed from $\hat{\alpha}$'s, $\hat{\beta}$'s, and θ's, we compute

$$\hat{\theta}_n = \begin{cases} \theta_n + \hat{\alpha}_n & \text{if } \theta_n + \hat{\alpha}_n \le \pi, \\ \theta_n + \hat{\alpha}_n - 2\pi & \text{if } \theta_n + \hat{\alpha}_n > \pi. \end{cases} \tag{3.7}$$

This formula ensures that $\hat{\theta}_n$ will satisfy $-\pi < \hat{\theta}_n \le \pi$ after the eigendecomposition of A is computed.

3.3. The Stopping Criterion. In practice a rootfinder can not be expected to make progress at a point λ where it is impossible to determine the sign of $\Phi(\theta)$. Motivated by [13], we use the following stopping criterion in the rootfinder:

$$\sum_{k=1}^{n} |z_k|^2 \left| \mathbf{fl}\left(\cot\left(\frac{\hat{\theta}_j - \theta_k}{2}\right)\right)\right| \le \eta \sum_{k=1}^{n} \frac{|z_k|^2}{\left|\mathbf{fl}\left(\sin((\hat{\theta}_j - \theta_k)/2)\right)\right|}, \tag{3.8}$$

where η is some appropriately chosen multiple of machine precision; and

$$\mathbf{fl}\left(\cot\left(\frac{\hat{\theta}_j - \theta_k}{2}\right)\right) \quad \text{and} \quad \mathbf{fl}\left(\sin((\hat{\theta}_j - \theta_k)/2)\right)$$

are the floating point results of computing the cot and sin functions by computing the arguments with the procedure described in § 3.2. Similar to [13], the right hand side in (3.8) is an upper bound on the round-off error in evaluating $\Phi(\hat{\theta}_j)$. Using arguments similar to those in [13], it can be shown that the set of approximate solutions satisfying (3.8) is non-empty in finite precision for any j. We would expect a good rootfinder to be able to compute such approximate solutions. In our FORTRAN implementation, we used a modified version of the rootfinder in [2, 11].

4. Numerical Stability of the Method. In this section we show that $\widehat{A}$ is close to A. Consider the following:

$$\begin{aligned} A - \widehat{A} &= \Lambda(I - 2zz^*) - \Lambda(I - 2\gamma\hat{z}\hat{z}^*) = 2\Lambda\left(\gamma\hat{z}\hat{z}^* - 2zz^*\right) \\ &= 2\Lambda\left((\gamma - 1)\hat{z}\hat{z}^* + (\hat{z} - z)\hat{z}^* - z(\hat{z} - z)^*\right). \end{aligned}$$

So, to show that $\widehat{A}$ is close to A, we only need to show that γ and $\hat{z}$ are close to 1 and z, respectively.

Before our formal analysis, we note that the secular equation (2.7) is derived under the condition that $\|z\|$ be exactly 1 (c.f. (2.4)), which rarely holds in practice. In addition, our analysis below will require that ν_1 and ν_n be computed to high relative accuracy, which may not be possible if ν_1 is close to $-\pi$ or ν_n close to π. To simplify the analysis, we assume for the moment that vector z in (2.4) satisfies $\|z\| = 1$ exactly and that scalars ν_1 and ν_n in (3.5) are known to high relative accuracy. We will come back to these assumptions at the end of § 4.

Under our assumption on the high relative accuracy in ν_1 and ν_n, the formulas established in § 3.2 for computing the sine function guarantee that we can compute $\sin((\theta_j - \theta_k)/2)$ and $\sin((\hat{\theta}_j - \theta_k)/2)$ to high relative accuracy for any j and k. Let us denote

$$d_{jk} = (\theta_j - \theta_k)/2, \;\; d_{jk}^{'} = (\theta_j^{'} - \theta_k)/2 \;\text{ and }\; \hat{d}_{jk} = (\hat{\theta}_j - \theta_k)/2.$$

Since $\Phi(\theta_j^{'}) = 0$, and

$$\sin\left((\hat{\theta}_j - \theta_j^{'})/2\right) = \sin(\hat{d}_{jk})\cos(d_{jk}^{'}) - \sin(d_{jk}^{'})\cos(\hat{d}_{jk}),$$

we have

$$-\Phi(\hat{\theta}_j) = \Phi(\theta_j^{'}) - \Phi(\hat{\theta}_j) = \sin\left((\hat{\theta}_j - \theta_j^{'})/2\right)\sum_{k=1}^{n}\frac{|z_k|^2}{\sin(d_{jk}^{'})\sin(\hat{d}_{jk})}.$$

The rootfinder guarantees that $\hat{\theta}_j$ and $\theta_j^{'}$ are in the same interval (θ_j, θ_{j+1}) for $j < n$, and that $\hat{\theta}_n$ and $\theta_n^{'}$ are in the same interval $(\theta_n, \theta_1 + 2\pi)$, which ensures that the product $\sin(\hat{d}_{jk})\sin(d_{jk}^{'})$ is always positive.

Combining the above equation with stopping criterion (3.8), we have

$$\begin{aligned} \left|\sin\left((\hat{\theta}_j - \theta_j^{'})/2\right)\right|\sum_{k=1}^{n}\frac{|z_k|^2}{|\sin(d_{jk}^{'})\sin(\hat{d}_{jk})|} &= \left|\sin\left((\hat{\theta}_j - \theta_j^{'})/2\right)\right|\left|\sum_{k=1}^{n}\frac{|z_k|^2}{\sin(d_{jk}^{'})\sin(\hat{d}_{jk})}\right| \\ &= |\Phi(\hat{\theta}_j)| \le \eta\sum_{k=1}^{n}\frac{|z_k|^2}{|\sin(\hat{d}_{jk})|} \le \eta\sum_{k=1}^{n}\frac{|z_k|^2}{|\sin(d_{jk}^{'})\sin(\hat{d}_{jk})|}. \end{aligned} \tag{4.1}$$

This yields the following result:

$$\left|\sin\left((\hat{\theta}_j - \theta_j^{'})/2\right)\right| \le \eta. \tag{4.2}$$

Hence

$$|\lambda_j' - \hat{\lambda}_j| = |\exp(\theta_j') - \exp(\hat{\theta}_j)| = 2\left|\sin\left((\hat{\theta}_j - \theta_j')/2\right)\right| \leq 2\eta, \tag{4.3}$$

which is to say that the eigenvalues are computed to full accuracy.

Note that the third condition in (2.9) guarantees that $\hat{\theta}_j - \theta_j'$ can not be too close to $\pm 2\pi$ for $j \leq n$:

$$\left|\left(\hat{\theta}_j - \theta_j'\right)/2\right| \leq \pi - \epsilon''/2.$$

We choose $\sin(\epsilon''/2) > \eta$. These two conditions guarantee that $\hat{\theta}_j$ and θ_j' in (4.2) must satisfy

$$\left|\hat{\theta}_j - \theta_j'\right| \leq 2\sin^{-1}\eta. \tag{4.4}$$

Now we use the above inequality to show that γ is close to 1. It follows from (2.25) and (2.28) that $\gamma = \exp\left(i\sum_{j=1}^{n}\frac{\hat{\theta}_j - \theta_j'}{2}\right)$. Combining this with (4.4),

$$\begin{aligned}|\gamma - 1| &= \left|\prod_{j=1}^{n}\exp\left(i\left(\frac{\hat{\theta}_j - \theta_j'}{2}\right)\right) - 1\right| \leq \prod_{j=1}^{n}\left(1 + \left|\exp\left(i\left(\frac{\hat{\theta}_j - \theta_j'}{2}\right)\right) - 1\right|\right) - 1\\ &= \prod_{j=1}^{n}\left(1 + 2\left|\sin\left(\frac{\hat{\theta}_j - \theta_j'}{4}\right)\right|\right) - 1 \leq \prod_{j=1}^{n}\left(1 + \left|\frac{\hat{\theta}_j - \theta_j'}{2}\right|\right) - 1\\ &\leq \left(1 + \sin^{-1}\eta\right)^n - 1 \leq e^{n\sin^{-1}\eta} - 1 \leq (e-1)n\sin^{-1}\eta, \qquad (4.5)\end{aligned}$$

where we have used the fact that $(e^x - 1)/x \leq e - 1$ for $0 \leq x \leq 1$.

To show that $\hat{z}$ is close to z, we need the following lemma.

LEMMA 4.1. *Let* $\sin y \neq 0$. *Then*

$$\sqrt{\left|\frac{\sin x}{\sin y}\right|} + \left|\frac{\sin(x \pm y)}{\sin y}\right| \geq \frac{2}{\pi} \quad \textit{and} \quad \sqrt{|\sin x\ \sin y|} \leq |\sin x| + |\sin(x \pm y)|. \tag{4.6}$$

Proof:. To prove the first inequality in (4.6), we first consider the case $0 \leq x \leq \pi/2$ and $0 < y \leq \pi/2$. Since it holds trivially if $x \geq y$, we further assume that $x < y$. Using (3.1), we get that

$$\begin{aligned}\sqrt{\left|\frac{\sin x}{\sin y}\right|} + \left|\frac{\sin(x \pm y)}{\sin y}\right| &\geq \sqrt{\frac{\sin x}{\sin y}} + \frac{\sin(y - x)}{\sin y} \geq 2/\pi\left(\sqrt{\frac{x}{y}} + \frac{y - x}{y}\right)\\ &= 2/\pi\left(1 + \sqrt{\frac{x}{y}}\left(1 - \sqrt{\frac{x}{y}}\right)\right) \geq 2/\pi.\end{aligned}$$

Hence the first relation in (4.6) holds when both $0 \leq x \leq \pi/2$ and $0 < y \leq \pi/2$. Replacing x by $\pi - x$ in the inequality, the resulting inequality is exactly the same, hence it holds when $\pi/2 < x \leq \pi$. Similarly, the inequality still holds when $\pi/2 \leq y < \pi$. Thus, It holds for any value of x and any $\sin y \neq 0$ due to periodicity.

To prove the second inequality in (4.6), we also restrict our attention to the special case $0 \le x < y \le \pi/2$. Let $a = y - x$. Then

$$\begin{aligned} |\sin x| + |\sin(x \pm y)| &\ge \sin x + \sin a \ge \sqrt{\sin^2 x + \sin x \sin a} \\ &\ge \sqrt{\sin^2 x \cos a + \sin x \sin a \cos x} = \sqrt{\sin x \sin(a + x)} = \sqrt{\sin x \sin y}. \end{aligned}$$

Hence the second inequality in (4.6) holds when $0 \le x < y \le \pi/2$. With arguments used earlier in the proof, it is straightforward to further conclude that this inequality holds for any x and y. ▮

Let $x = \hat{d}_{jk}$ and $y = d'_{jk}$ in (4.6), and we have

$$\frac{1}{|\sin(\hat{d}_{jk})|} \le \frac{\pi/2}{\sqrt{|\sin(\hat{d}_{jk}) \sin(d'_{jk})|}} + \frac{\pi/2\, |\sin(\hat{d}_{jk} - d'_{jk})|}{|\sin(\hat{d}_{jk}) \sin(d'_{jk})|}.$$

Note that $\hat{d}_{jk} - d'_{jk} = (\hat{\theta}_j - \theta'_j)/2$. Plugging the above into the first inequality in (4.1) and simplifying, we have

$$\begin{aligned} \left|\sin\left((\hat{\theta}_j - \theta'_j)/2\right)\right| \sum_{k=1}^{n} \frac{|z_k|^2}{|\sin(d'_{jk}) \sin(\hat{d}_{jk})|} &\le \frac{\pi\,\eta/2}{1 - \pi\,\eta/2} \sum_{k=1}^{n} \frac{|z_k|^2}{\sqrt{|\sin(d'_{jk}) \sin(\hat{d}_{jk})|}} \\ &\le \frac{\|z\|\,\pi\,\eta/2}{1 - \pi\,\eta/2} \sqrt{\sum_{k=1}^{n} \frac{|z_k|^2}{|\sin(d'_{jk}) \sin(\hat{d}_{jk})|}}, \end{aligned}$$

where we have used the Cauchy-Schwartz inequality. Further simplifying,

$$\begin{aligned} \left|\sin\left((\hat{\theta}_j - \theta'_j)/2\right)\right| &\le \frac{\|z\|\,\pi\,\eta/2}{1 - \pi\,\eta/2} \Bigg/ \sqrt{\sum_{k=1}^{n} \frac{|z_k|^2}{|\sin(d'_{jk}) \sin(\hat{d}_{jk})|}} \\ &\le \frac{\|z\|\,\pi\,\eta/2}{(1 - \pi\,\eta/2)|z_k|} \sqrt{|\sin(d'_{jk}) \sin(\hat{d}_{jk})|}. \qquad (4.7) \end{aligned}$$

Set $x = \hat{d}_{jk}$ and $y = d'_{jk}$ in the second inequality in (4.6), we have

$$\begin{aligned} \sqrt{|\sin(d'_{jk}) \sin(\hat{d}_{jk})|} &\le |\sin(d'_{jk})| + |\sin(\hat{d}_{jk} - d'_{jk})| \\ &= |\sin(d'_{jk})| + |\sin((\hat{\theta}_j - \theta'_j)/2)|. \end{aligned}$$

Plugging this into (4.7) and simplifying,

$$(4.8) \qquad \left|\sin\left((\hat{\theta}_j - \theta'_j)/2\right)\right| \le \frac{\|z\|\,\pi\,\eta/2}{(1 - \pi\,\eta/2)|z_k| - \|z\|\,\pi\,\eta/2} |\sin(d'_{jk})|.$$

Similar to (4.4), and in light of (3.1), we get from (4.8) that

$$\begin{aligned} \left|\hat{\theta}_j - \theta'_j\right| &\le 2 \sin^{-1}\left(\frac{\|z\|\,\pi\,\eta/2}{(1 - \pi\,\eta/2)|z_k| - \|z\|\,\pi\,\eta/2} |\sin(d'_{jk})|\right) \\ &\le \frac{\|z\|\,\pi^2\,\eta/2}{(1 - \pi\,\eta/2)|z_k| - \|z\|\,\pi\,\eta/2} |\sin(d'_{jk})| \overset{\text{def}}{=} \frac{2\delta_k\,\eta}{|z_k|} |\sin(d'_{jk})|. \qquad (4.9) \end{aligned}$$

In (2.9), we choose the deflation tolerance

$$\epsilon^{'} \geq \pi^2 n\eta/4 \geq \pi\eta/(1-\pi\eta).$$

This implies that

$$\delta_k = \frac{\|z\|\,\pi^2\,/4}{(1-\pi\,\eta/2) - \|z\|\,\pi\,\eta/(2|z_k|)} \leq \pi^2/2.$$

We are now in a position to show that $\hat{z}$ is close to z. Using (2.25), (2.26) and (2.29),

$$\begin{aligned} |\hat{z}_k - z_k| &= \left| |\hat{z}_k| \frac{z_k}{|z_k|} - |z_k| \frac{z_k}{|z_k|} \right| = ||\hat{z}_k| - |z_k|| \\ &= \left| \left| \frac{\prod_{j=1}^n \sin(\hat{d}_{jk})}{\prod_{j\neq k} \sin(d_{jk})} \right|^{1/2} - \left| \frac{\prod_{j=1}^n \sin(d^{'}_{jk})}{\prod_{j\neq k} \sin(d_{jk})} \right|^{1/2} \right| \\ &= |z_k| \left| \left| \prod_{j=1}^{n} \frac{\sin(\hat{d}_{jk})}{\sin(d^{'}_{jk})} \right|^{1/2} - 1 \right|. \end{aligned} \tag{4.10}$$

We seek bounds on each factor of the product. Using (3.1) and the identity,

$$\sin(x+y) - \sin(x-y) = 2\sin(y)\cos(x),$$

we get:

$$\left| \frac{\sin(\hat{d}_{jk})}{\sin(d^{'}_{jk})} - 1 \right| = \left| \frac{2\sin(\frac{\hat{\theta}_j - \theta^{'}_j}{4})\ \cos(\frac{\hat{\theta}_j + \theta^{'}_j}{4} - \frac{\theta_k}{2})}{\sin(d^{'}_{jk})} \right| \leq \left| \frac{\frac{\hat{\theta}_j - \theta^{'}_j}{2}}{\sin(d^{'}_{jk})} \right|.$$

With (4.9) and the upper bound on δ_k, we have

$$\left| \frac{\sin(\hat{d}_{jk})}{\sin(d^{'}_{jk})} - 1 \right| \leq \frac{\delta_k\,\eta}{|z_k|} \leq \frac{\pi^2\,\eta}{2|z_k|}. \tag{4.11}$$

Plugging this into (4.10),

$$|\hat{z}_k - z_k| \leq |z_k| \left(\left(1 + \frac{\pi^2\,\eta}{2|z_k|}\right)^{n/2} - 1 \right) \leq |z_k| \left(e^{\pi^2 n\,\eta/(4|z_k|)} - 1 \right).$$

Using the fact that $\pi^2 n\eta/(4|z_k|) \leq 1$ and that $(e^x - 1)/x \leq e - 1$ for $0 \leq x \leq 1$,

$$|\hat{z}_k - z_k| \leq |z_k|(e-1)\pi^2 n\eta/(4|z_k|) \leq \pi^2 n\eta/2. \tag{4.12}$$

This last relation implies that $\hat{z}$ is indeed close to z. And hence we conclude that $\widehat{A}$ is indeed close to A.

Finally, we address the issues regarding $\|z\|$ and the computed accuracy in ν_1 and ν_n. We note that since $\|z\|$ is close to 1, the matrix

$$\widetilde{A} \stackrel{\text{def}}{=} \Lambda\,(I - \tilde{z}\,\tilde{z}^*)\,, \quad \tilde{z} = z/\|z\|,$$

is close to A with $\|\tilde{z}\| = 1$. The stopping criteria (3.8) for $\widetilde{A}$ and A differ by a common factor $1/\|z\|$ on both sides of the inequality and hence are equivalent. Repeating the above analysis leading to (4.12), we conclude that $\widehat{A}$ is close to $\widetilde{A}$, and thus to A. We point out that $\widetilde{A}$ is constructed for the above analysis only, not actual computation.

We have also developed a somewhat more detailed analysis paralleling the one in this section to show that $\widehat{A}$ is still close to $\widetilde{A}$ even if ν_1 and ν_n are computed only to high absolute accuracy. We omit it in our paper for the following reasons: first, this analysis is quite technical and does not provide additional insight; and second, ν_1 and ν_n can easily be computed to high relative accuracy with emulated extra precision techniques (see Priest [15]).

5. Numerical Experiments. We now present some experimental results to compare the performance of our method against the "old UDC" described in [2] and the HQR methods in [5]. To make easy comparison with the FORTRAN subroutines UDC and CHSEQR, we have implemented our algorithm in FORTRAN as well. Below are four graphs representing the performance of the three algorithms. All three were run on a Sparc-20 workstation in single precision arithmetic, roughly corresponding to seven significant digits. Deflation tolerance was set to 10^{-6}.

FIG. 5.1. *Efficiency of Method*

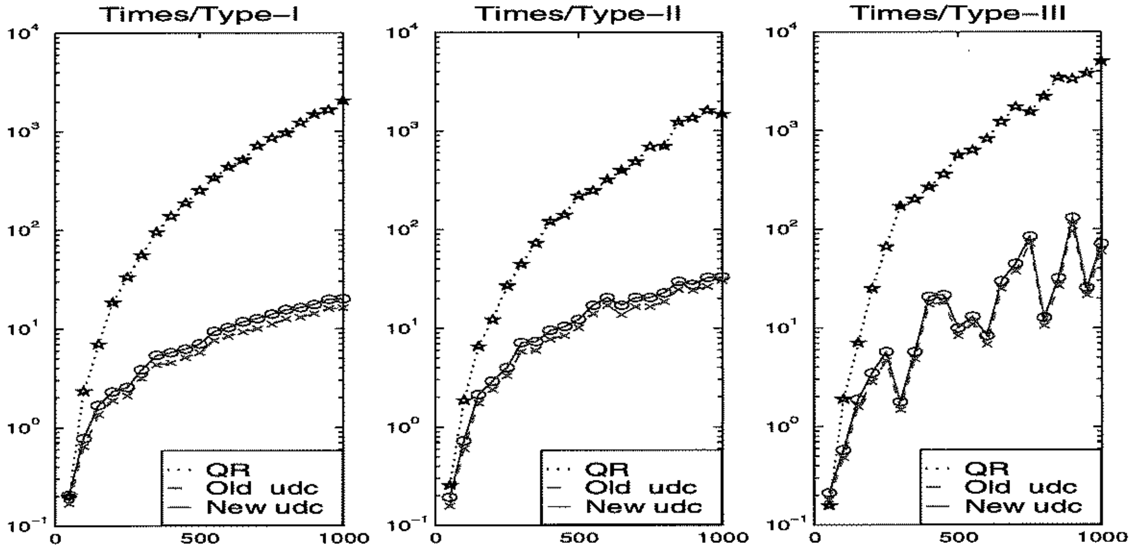


FIG. 5.2. *Fitting Efficiency with Quadratic Least Squares*

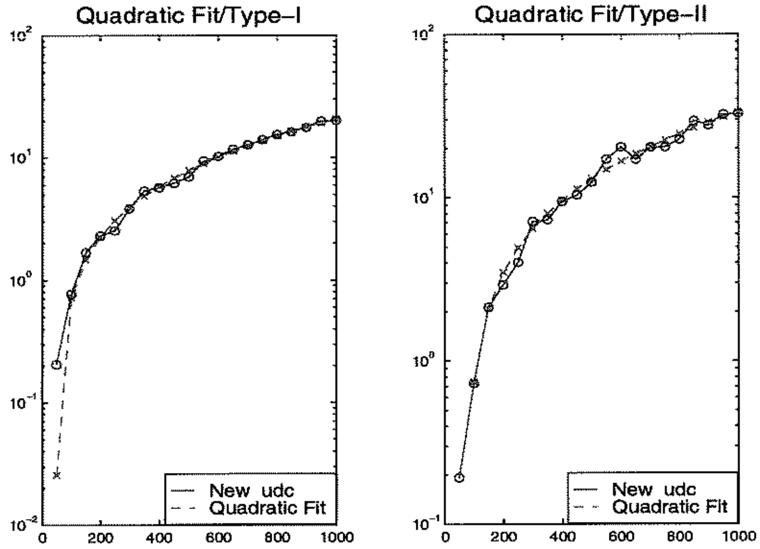
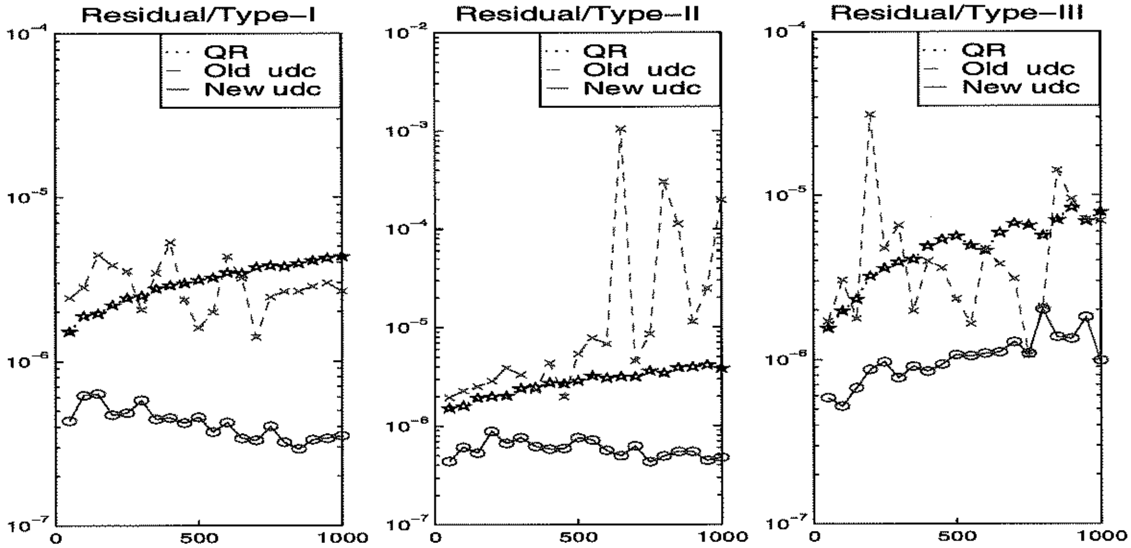
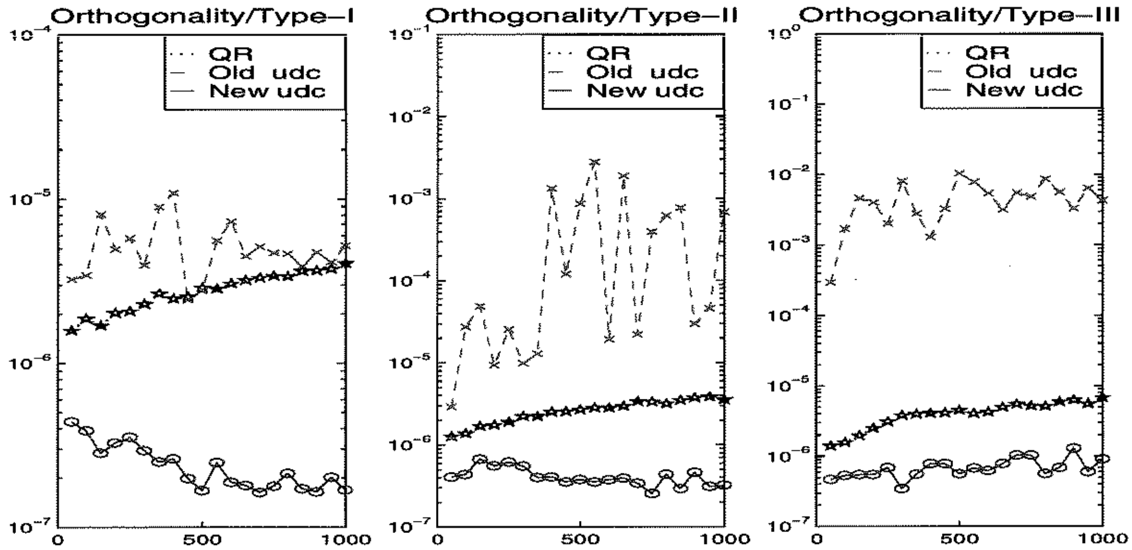


FIG. 5.3. $\|HW - W\Omega\|_\infty/\sqrt{n}$



We considered twenty matrices ranging from size 50 to size 1000, measuring the speed of the algorithms, the accuracy of the spectral resolution compared to the original matrix, and the orthogonality of the eigenvectors. For Figure 5.1, we measured the speed of the algorithms in seconds. To calculate how close the computed spectral resolution came to the approximating H, we took the infinity norm of $H\,W - W\,\Omega$. If all arithmetic was done in exact precision, this residual should equal zero. Figure 5.3 illustrates the numerical value of the residual. Similarly, to calculate how close the eigenvectors come to being orthogonal, we took the infinity norm of $W^*W - I$. Figure 5.4 illustrates this numerical value.

FIG. 5.4. $\|W^*W - I\|_\infty/\sqrt{n}$



We experimented on three kinds of matrices. In Figures 5.1, 5.3 and 5.4, the dotted line marked off with asterisks represents the performance of the HQR from LAPACK; the dashed line marked off with x'es represents the performance of the "old UDC" code; and the solid line marked off with o's represents the performance of our method, the "new UDC" code.

- In our first experiment, we simply considered randomly generated unitary upper Hessenberg matrices. Such an H is constructed by inputting the Schur parameters, $\gamma_j = \rho_j \exp(i\alpha_j), 1 \leq j \leq n$, where the α_j are uniformly distributed random variables on $[0, 2\pi]$ and the ρ_j are uniformly distributed on $[0,1]$ and $\rho_n = 1$. This Schur decomposition ensures that H is unitary.
  The results of this experiment showed that our method improved upon the original UDC method by roughly a factor of 10 or more with regard to both the residual and orthogonality of the eigenvectors. It is also much faster and significantly more accurate than the HQR code. The original UDC performs only slightly faster then our method. Additionally, the speed of our method seemed to be on the order of the square of the size of the matrix, since the data seems to fit a quadratic polynomial of n quite well (See Figure 5.2).
- In our next experiment, we considered matrices which have one or more eigenvalues whose arguments are near $\pm\pi$. This is constructed by creating a real-valued matrix H with odd size. Then, one of the eigenvalues cannot have a distinct complex conjugate, thereby forcing that eigenvalue to equal 1 or -1. By setting $\gamma_n = -1$, we force the real eigenvalue to have an argument at π.
  The results of this experiment showed remarkable improvement on the original UDC algorithm. For sizable matrices, the original method becomes highly unstable, producing inaccurate results. An examination of the results for a matrix of size 651 reveals the residual and orthogonality results on the "old UDC" to be somewhere in the neighborhood of 10^{-3}, whereas the results for our method stayed stable around 10^{-6}, the deflation tolerance. Again, our method compares very favorably with the HQR code. Similar to the previous

experiment, the efficiency of the new method is only slightly worse than the original method, and Figure 5.2 still suggests that the speed of our UDC is quadratic with respect to the size of the problem.

- In the third experiment, we designed H to have nearly multiple eigenvalues. We do this by making H nearly block diagonal with identical blocks. As described in [2], we let $n = pk$. Generate the first $p-1$ Schur parameters as in the first experiment. Then set σ_p equal to some small constant. The remaining parameters are given by $\gamma_{lp+j} = \gamma_j, \sigma_{lp+j} = \sigma_j, 1 \leq j \leq p, 1 \leq l < k$. Then set $\gamma_n = 1$. If $\sigma_p = 0$, then the eigenvalues of H occur with multiplicity k. Otherwise for small σ_p, we get nearly multiple eigenvalues.
  Experimental results on the third experiment once again show a vast improvement over the original UDC method and HQR, with regards to stability of the eigenvector calculations. Figure 5.1 indicates that the efficiency effect of deflation on Type-III matrices is both dramatic and erratic, making predicting the speed of our UDC for Type-III matrices difficult.

6. Conclusion. This paper has outlined a stable algorithm for computing the spectral resolution of a unitary upper Hessenberg matrix. We showed that our algorithm is stable regardless of eigenvalue distribution of the given problem. The computed eigenvalues are all unit modulus, and the computed eigenvectors are all numerically orthogonal.

This method relied on several delicate techniques. First, as in all Divide and Conquer methods, we required a deflation procedure to ensure that we could find the roots of the spectral function. Additionally, in the calculation of the eigenvectors of H, special attention is given to the way that angles are handled. Finally, we used a matrix reconstruction idea from [13, 14] to guarantee that the computed eigenvectors are automatically orthogonal.

Future work includes parallelization of the new UDC algorithm and developing a simplified version for the special case where the input data is all real.

REFERENCES

[1] G. S. Ammar, W. B. Gragg, L. Reichel, *Determination of Pisarenko Frequency Estimates as Eigenvalues of an Orthogonal Matrix, In F.T. Luk, ed., Advanced Algorithms and Architectures for Signal Processing II, Proc. SPIE 826*, (1987), pp. 143–145.

[2] G. S. Ammar, L. Reichel, D.C. Sorensen, *An Implementation of a Divide and Conquer Algorithm for the Unitary Eigenproblem, ACM Transactions on Mathematical Software*, vol. 18 (1992), pp. 292–307.

[3] G. S. Ammar, W. B. Gragg, L. Reichel, *Direct and Inverse Unitary Eigenvalue Problems in Signal Processing: An Overview, In M.S. Moonen, G.H. Golub and B.L.R. De Moor, eds., Linear Algebra for Large Scale and Real Time Applications*, NATO ASI Series (1993), pp. 341–343.

[4] G. S. Ammar, *The QR Algorithm for Orthogonal Hessenberg Matrices, 7th Conference of the International Linear Algebra Society*, (1998).

[5] 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*, SIAM, Philadelphia, PA, third ed., 1999.

[6] J. J. M. Cuppen, *A Divide and Conquer Method for the Symmetric Tridiagonal Eigenproblem, Numer. Math.*, vol. 36 (1981), pp. 177–195.

[7] J.J. Dongarra and D.C. Sorensen, *A Fully Parallel Algorithm for the Symmetric Eigenvalue Problem, SIAM J. Sci. Stat. Comput.*, vol. 8 (1987), pp. s139–s154.

[8] G. Golub and C. Van Loan, *Matrix Computations, Johns Hopkins University Press*, 3rd ed (1996).

[9] W. B. Gragg, *The QR Algorithm for Unitary Hessenberg Matrices, Journal of Computational and Applied Mathematics*, (1986).

[10] W. B. Gragg and L. Reichel, *A Divide and Conquer Algorithm for the Unitary Eigenproblem, In M.T. Heath, ed., Hypercube Multiprocessors*, (1987), pp. 639–647.

[11] W. B. Gragg and L. Reichel, *A Divide and Conquer Method for Unitary and Orthogonal Eigenproblems, Numer. Math.*, vol. 57 (1990), pp. 695–718.

[12] W. B. Gragg, *Stabilization of the uhqr Algorithm, 7th Conference of the International Linear Algebra Society*, (1998).

[13] M. Gu and S. C. Eisenstat, *A Stable and Efficient Algorithm for the Rank-One Modification of the Symmetric Eigenproblem, SIAM J. Matrix Anal. Appl.*, vol. 15 (1994), pp. 1266–1276.

[14] M. Gu and S. C. Eisenstat, *A Divide-and-Conquer Algorithm for the Symmetric Tridiagonal Eigenproblem, SIAM J. Matrix Anal. Appl.*, vol. 16 (1995), pp. 79–92.

[15] D. Priest, *Algorithms for arbitrary precision floating point arithmetic*, in Proceedings of the 10th Symposium on Computer Arithmetic, P. Kornerup and D. Matula, eds., Grenoble, France, June 26-28 1991, IEEE Computer Society Press, pp. 132–145.

[16] M. Stewart, *An Error Analysis of a Unitary Hessenberg QR Algorithm, Australian Nat'l University, Joint CS Technical Report Series*, (1998), pp. 17–29.

[17] J.-G. Sun, *Residual bounds on approximate solutions for the unitary eigenproblem*, SIAM J. Matrix Anal. Appl., 17 (1996), pp. 69–82.