Electronic Transactions on Numerical Analysis.
Volume 4, pp. 64-74, June 1996.

ISSN 1068-9613.

ETNA
Kent State University
etna@mcs.kent.edu

A NOTE ON NEWBERY'S ALGORITHM FOR DISCRETE LEAST-SQUARES APPROXIMATION BY TRIGONOMETRIC POLYNOMIALS*

HEIKE FAßBENDER†

Abstract. Recently fast, efficient and reliable algorithms for discrete least-squares approximation of a real-valued function given at arbitrary distinct nodes in $[0, 2\pi)$ by trigonometric polynomials were presented in different papers. These algorithms are based on schemes for the solution of inverse unitary eigenproblems and require only $\mathrm{O}(mn)$ arithmetic operations as compared to $\mathrm{O}(mn^2)$ operations needed for algorithms that ignore the structure of the problem. In 1970 Newbery already presented a $\mathrm{O}(mn)$ algorithm for solving the discrete least-squares approximation by trigonometric polynomials. In this paper the connection between the different algorithms is illustrated.

Key words. trigonometric approximation.

AMS subject classifications. 65D10, 42A10, 65F99.

1. Introduction.

A problem in signal processing is the approximation of a function known only at some measured points by a trigonometric function. A number of different models for representing the measured points as a finite superposition of sine- and cosine-oscillations are possible. One choice could be to compute the trigonometric interpolating function. Then several numerical algorithms are available ([3, 4, 9]). But in general a large number of measured points are given, so that this approach leads to a trigonometric polynomial with a lot of superimposed oscillations (and a large linear system to solve). In practical applications it is often sufficient to compute a trigonometric polynomial with only a small number of superimposed oscillations. A different, often chosen approach is the (fast) Fourier transform ([9]). In this case the frequencies of the sine- and cosine-oscillations have to be chosen equidistant. More freedom in the choice of the frequencies and the number of superimposed oscillations is given in the following approach: Given a set of m arbitrary distinct nodes $\{\theta_k\}_{k=1}^m$ in the interval $[0, 2\pi)$, a set of m positive weights $\{\omega_k^2\}_{k=1}^m$, and a real-valued function $f(\theta)$ whose values at the nodes θ_k are explicitly known, then the trigonometric function

$$t(\theta) = a_0 + \sum_{j=1}^{\ell} (a_j \cos j\theta + b_j \sin j\theta), \qquad a_j, b_j \in \mathbb{R} \tag{1.1}$$

of order at most $\ell < m/2$ is sought that minimizes the discrete least-squares error

$$||f - t||_{\mathbb{R}} := \sqrt{\sum_{k=1}^{m} |f(\theta_k) - t(\theta_k)|^2 \omega_k^2}. \tag{1.2}$$

In general, m (the number of measured functional values) is much larger than $n = 2\ell + 1$ (the number of coefficients to be determined).

Standard algorithms for solving the approximation problem (1.2) require $\mathrm{O}(mn^2)$ arithmetic operations. Recently this problem received attention as it was noted that

* Received January 18, 1996. Accepted for publication June 25, 1996. Communicated by G. S. Ammar.
† Universität Bremen, Fachbereich 3 Mathematik und Informatik, 28334 Bremen, Germany (heike@mathematik.uni-bremen.de).

ETNA
Kent State University
etna@mcs.kent.edu

fast algorithms can be developed [11, 5, 7]. These algorithms are based on schemes for the solution of inverse unitary eigenvalue problems and require merely $O(mn)$ arithmetic operations. Section 2 gives a short description of these algorithms. Already in 1970, Newbery [10] presented a $O(mn)$ algorithm to solve the least-squares approximation by trigonometric polynomials. This algorithm is closely related to the algorithms described in Section 2. In Section 3 Newbery's algorithm is given. Its connection to the algorithms given in Section 2 is described in Section 4.

2. Algorithms based on the inverse unitary eigenproblem. In [11] Reichel, Ammar, and Gragg reformulate the problem (1.2) as the following standard least-squares problem: Minimize

$$||DAc - Dg||_2 = \min, \tag{2.1}$$

where $D = diag(\omega_1, ..., \omega_m) \in \mathbb{C}^{m\times m}$ is a diagonal matrix with the given weights on the diagonal and A is a transposed Vandermonde matrix

$$A = \begin{pmatrix} 1 & z_1 & \cdots & z_1^{n-1} \\ 1 & z_2 & \cdots & z_2^{n-1} \\ \vdots & \vdots & & \vdots \\ 1 & z_m & \cdots & z_m^{n-1} \end{pmatrix} \in \mathbb{C}^{m\times n}$$

with $z_k = exp(i\theta_k)$. $g = [g(z_1), ..., g(z_m)]^T \in \mathbb{C}^m$ is a vector of the values of a complex function $g(z)$ and $c = [c_0, ..., c_{n-1}]^T \in \mathbb{C}^n$ is the solution vector. With the proper choice of n and g, it is easy to see that the coefficients of the trigonometric polynomial (1.1) that minimizes the error (1.2) can be read off of the least-squares solution $\widehat{c}$ of (2.1) (see [11]).

The solution $\widehat{c}$ can be computed by using the QR decomposition of DA. Since DA has full column rank, there is an $m \times m$ unitary matrix Q with orthonormal columns and an $m \times n$ upper triangular matrix R with positive diagonal elements such that

$$DA = QR = (Q_1|Q_2)\begin{pmatrix} R_1 \\ 0 \end{pmatrix} = Q_1R_1,$$

where $Q_1 \in \mathbb{C}^{m\times n}$ has orthonormal columns and $R_1 \in \mathbb{C}^{n\times n}$ has positive diagonal elements. The solution of (2.1) is given by $\widehat{c} = R_1^{-1}Q_1^HDg$. Algorithms that solve the least squares problem via the QR decomposition of DA without using the special structure require $O(mn^2)$ arithmetic operations ([8]).

In [11] Reichel, Ammar, and Gragg present an approach to compute the QR decomposition of DA that is based on computational aspects associated with the family of polynomials orthogonal with respect to an inner product on the unit circle. Such polynomials are known as Szegő polynomials. The following interpretation of the elements of Q_1 and R_1 in terms of Szegő polynomials can be given : Q_1 is determined by the values of the Szegő polynomials at the nodes z_k. R_1 expresses the power basis in terms of the orthonormal Szegő polynomials. Therefore, the columns of R_1^{-1} are the coefficients of the Szegő polynomials in the power basis. There exist algorithms for determining the values of the Szegő polynomials at nodes z_k ([11, 6]) which require $O(mn)$ arithmetic operations. The computation of the columns of R_1^{-1} relies on the Szegő recursion and is closely related to the Levinson algorithm as $(DA)^TDA = R_1^TR_1$ is a Toeplitz matrix.

ETNA
Kent State University
etna@mcs.kent.edu

Observe that

$$
\begin{aligned}
DA &= \begin{pmatrix} \omega_1 & \omega_1 z_1 & \omega_1 z_1^2 & \cdots & \omega_1 z_1^{n-1} \\ \vdots & \vdots & \vdots & & \vdots \\ \omega_m & \omega_m z_m & \omega_m z_m^2 & \cdots & \omega_m z_m^{n-1} \end{pmatrix} \\
&= (q, \Lambda q, \Lambda^2 q, ..., \Lambda^{n-1} q) \\
&= \sigma_0 (q_0, \Lambda q_0, \Lambda^2 q_0, ..., \Lambda^{n-1} q_0)
\end{aligned}
$$

with $q := (\omega_1, ..., \omega_m)^T, \sigma_0 = ||q||_2, q_0 := \sigma_0^{-1} q$ and $\Lambda = diag(z_1, ..., z_m)$. Thus, the matrix DA is given by the first n columns of the Krylov matrix $K(\Lambda, q, m) = (q, \Lambda q, ..., \Lambda^{m-1} q)$. We may therefore use the following consequence of the Implicit Q Theorem (see, for instance, [8, p. 367]) to compute the desired QR decomposition: If there exists a unitary matrix U such that $U^H \Lambda U = H$ is a unitary upper Hessenberg matrix with positive subdiagonal elements, then the QR decomposition of $K(\Lambda, q_0, m)$ is given by UR with $R = K(H, e_1, m)$. The construction of such a unitary Hessenberg matrix from spectral data, here-contained in Λ and q_0, is an inverse eigenproblem. Thus the best trigonometric approximation to f can be computed via solving this inverse eigenproblem. Because of the uniqueness of the here-given QR decomposition of $K(\Lambda, q_0, m)$, it follows from the above given interpretation of the elements of Q_1 that the elements in U are the values of the Szegő polynomials at the nodes z_k. Thus solving the inverse unitary Hessenberg eigenvalue problem $U^H \Lambda U = H$ is equivalent to computing values of Szegő polynomials.

Unitary Hessenberg matrices have special properties which allow the development of efficient algorithms for this class of matrices. Any $n \times n$ unitary Hessenberg matrix with positive subdiagonal elements can be uniquely parametrized by n complex parameters, that is

$$H = G_1(\gamma_1) G_2(\gamma_2) \cdots G_n(\gamma_n),$$

for certain complex-valued parameters $|\gamma_k| < 1, 1 \leq k < n$, and $|\gamma_n| = 1$. Here $G_k(\gamma_k)$ denotes the $n \times n$ Givens reflector in the $(k, k+1)$ plane

$$G_k = G_k(\gamma_k) = diag(I_{k-1}, \begin{bmatrix} -\gamma_k & \sigma_k \\ \sigma_k & \overline{\gamma_k} \end{bmatrix}, I_{n-k-1})$$

with $\gamma_k \in \mathbb{C}$, $\sigma_k \in \mathbb{R}^+$, $|\gamma_k|^2 + \sigma_k^2 = 1$, and

$$G_n(\gamma_n) = diag(I_{n-1}, -\gamma_n)$$

with $\gamma_n \in \mathbb{C}$, $|\gamma_n| = 1$. The nontrivial entries γ_k are called *Schur parameters* and the σ_k are called *complementary Schur parameters*. Ammar, Gragg, and Reichel make use of this parametrization in [2] by developing an efficient and reliable algorithm (IUQR-algorithm) for solving the inverse unitary Hessenberg eigenvalue problem. The algorithm manipulates the n complex parameters instead of the n^2 matrix elements. An adaptations of the IUQR scheme to the computation of the vector $c' = Q_1^H D g$ can be given, which requires $O(mn)$ arithmetic operations. After computing the vector c', the least-squares solution $\widehat{c} = R_1^{-1} c'$ of (2.1) can be obtained using an algorithm closely related to the Levinson algorithm. Reichel, Ammar, and Gragg present in [11] a $O(n^2)$ algorithm to compute $R_1^{-1} b$ for an arbitrary vector $b \in \mathbb{C}^n$.

The algorithms proposed by Reichel, Ammar, and Gragg in [11] construct the least-squares solution $\widehat{c}$ of (2.1) in $O(mn + n^2)$ arithmetic operations. The coefficients of the optimal trigonometric polynomial t of (1.2) can be recovered from $\widehat{c}$. This representation of t is convenient if we desire to integrate or differentiate the polynomial or if we wish to evaluate it at many equidistant points on a circle with a center at the origin. If we, on the other hand, only desire to evaluate t at a few points, then we can use the representation of t in terms of Szegő polynomials. For details see [11].

The method proposed by Reichel, Ammar, and Gragg to solve the real-valued approximation problem (1.2) computes the real-valued solution using complex arithmetic by solving an inverse unitary Hessenberg eigenvalue problem $U^H \Lambda U = H$, where a unitary Hessenberg matrix is constructed from spectral data. Now $H = G_1(\gamma_1)G_2(\gamma_2)\cdots G_n(\gamma_n)$ can be transformed to $G_oG_e^H$ by a unitary similarity transformation (see [1]), where

$$G_o = G_1(\gamma_1)G_3(\gamma_3)\cdots G_{2[(n+1)/2]-1}(\gamma_{2[(n+1)/2]-1}) = \begin{pmatrix} -\gamma_1 & \sigma_1 & & & \\ \sigma_1 & \overline{\gamma}_1 & & -\gamma_3 & \sigma_3 \\ & & \sigma_3 & \overline{\gamma}_3 & \\ & & & & \ddots \end{pmatrix}$$

is the product of the odd numbered elementary reflectors and

$$G_e^H = G_2(\gamma_2)G_4(\gamma_4)\cdots G_{2[n/2]}(\gamma_{2[n/2]}) = \begin{pmatrix} 1 & & & \\ & -\gamma_2 & \sigma_2 & \\ & \sigma_2 & \overline{\gamma}_2 & \\ & & & \ddots \end{pmatrix}$$

is the product of the even numbered elementary reflectors. Here $[x] = \max\{i \in \mathbb{N} | i \leq x\}$. G_o, G_e are block diagonal matrices with block size at most two. Thus the inverse unitary Hessenberg eigenvalue problem $U^H\Lambda U = H$ is equivalent to an inverse eigenvalue problem $Q^H(\Lambda - \lambda I)QG_e = G_o - \lambda G_e$, where a *Schur parameter pencil* is constructed from spectral data.

In [5, 7] numerical methods for the trigonometric approximation are discussed which rely on this inverse eigenvalue problem for Schur parameter pencils. Especially, an algorithm is developed which requires $\mathrm{O}(mn)$ arithmetic operations to solve the real-valued approximation problem (1.2) using only real arithmetic. The following approach for solving the approximation problem (1.2) is considered: Since

$$\underbrace{\begin{pmatrix} 1 & \sin\theta_1 & \cos\theta_1 & \cdots & \sin\ell\theta_1 & \cos\ell\theta_1 \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ 1 & \sin\theta_m & \cos\theta_m & \cdots & \sin\ell\theta_m & \cos\ell\theta_m \end{pmatrix}}_{\widetilde{A}} \underbrace{\begin{pmatrix} a_0 \\ b_1 \\ a_1 \\ \vdots \\ b_\ell \\ a_\ell \end{pmatrix}}_{\widetilde{t}} = \underbrace{\begin{pmatrix} t(\theta_1) \\ \vdots \\ t(\theta_m) \end{pmatrix}}_{\widehat{t},}$$

it follows with $D = diag(\omega_1, ..., \omega_m)$ and $\widehat{f} = (f(\theta_1), ..., f(\theta_m))^T$ that (1.2) can be reformulated to the real-valued least-squares problem

$$||f - t||_{\mathbb{R}} = ||D(\widehat{f} - \widehat{t})||_2 = ||D\widehat{f} - D\widetilde{A}\widetilde{t}||_2. \tag{2.2}$$

This least-squares problem can be solved via QR decomposition of $D\widetilde{A}$. As $D\widetilde{A}$ is a real $m \times n$ matrix with full column rank, there exists a unique "skinny" real

QR decomposition $\widetilde{Q}_1\widetilde{R}_1$ of $D\widetilde{A}$ where $\widetilde{Q}_1 \in \mathbb{R}^{m\times n}$ has orthonormal columns and $\widetilde{R}_1 \in \mathbb{R}^{n\times n}$ is upper triangular with positive diagonal entries [8, Theorem 5.2.2]. The solution of (2.2) is thus given by $\widetilde{t} = \widetilde{R}_1^{-1}\widetilde{Q}_1^H D\widehat{f}$.

In [5, 7] an approach to compute the QR decomposition is presented that is based on computational aspects associated with a family of Laurent polynomials orthogonal with respect to an inner product on the unit circle. These orthogonal Laurent polynomials are closely related to the Szegő polynomials (see [5, 6]).

Observe that

$$D\widetilde{A} = \frac{1}{2}\kappa(\Lambda, q, \ell)F$$

with $\Lambda and q$ as before, and with

$$\begin{aligned} \kappa(\Lambda, q, \ell) &= [q, \Lambda q, \Lambda^H q, \Lambda^2 q, (\Lambda^H)^2 q, ..., \Lambda^\ell q, (\Lambda^H)^\ell q] \in \mathbb{C}^{m\times(2\ell+1)} \\ \text{and } F &= diag(2, \begin{bmatrix} -i & 1 \\ i & 1 \end{bmatrix}, ... \begin{bmatrix} -i & 1 \\ i & 1 \end{bmatrix}). \end{aligned}$$

The following consequence of the Implicit Q Theorem can be used to compute the QR decomposition of $\kappa(\Lambda, q, \ell)$: If there exist unitary matrices Q and P such that $Q(\Lambda - \lambda I)P = G_o - \lambda G_e$ is an unreduced Schur parameter pencil (with positive complementary Schur parameters), then the QR decomposition of $\kappa(\Lambda, q, \ell)$ is given by $\sigma_0 QR$ with $R = \kappa(G_o G_e^H, e_1, \ell)$. The construction of such a Schur parameter pencil from spectral data, here contained in Λ and q, is an inverse eigenproblem. Thus the best trigonometric approximation to f can be computed via solving this inverse eigenproblem. The elements of Q can be interpreted as the values of Laurent polynomials orthogonal with respect to an inner product on the unit circle. Solving the inverse eigenvalue problem $Q(\Lambda - \lambda I)P = G_o - \lambda G_e$ is equivalent to computing values of orthogonal Laurent polynomials. Making use of the parametrization of G_o, G_e^H an efficient and reliable algorithm for solving the inverse eigenvalue problem for Schur parameter pencils is developed in [5, 7]. An adaption of this scheme to the computation of the vector $t' = Q_1^H D\widehat{f}$ can be given, which requires $O(mn)$ arithmetic operations (Q_1 denotes again the first n columns of Q). The computation of R implicitly yields the Cholesky factorization of a bordered block-Toeplitz-plus-block-Hankel matrix with 2×2 blocks. After computing the vector t', the least-squares solution $\widetilde{t} = 2\sigma_0^{-1}F^{-1}R_1^{-1}t'$ of (2.2) can therefore be obtained using an algorithm closely related to the Levinson algorithm. Algorithms for inverting the upper square subblock of R which require only $O(n^2)$ arithmetic operations are given in [5, 7]. Thus the least-squares solution $\widetilde{t}$ of (2.2) can be constructed in $O(mn + n^2)$ arithmetic operations.

Finally, in [5, 7] algorithms for computing the unique "skinny" real-valued QR decomposition $D\widetilde{A} = \widetilde{Q}_1\widetilde{R}_1$ are developed, which use only real arithmetic and require merely $O(mn)$ arithmetic operations. For that purpose the effect of the transformation matrix $\widetilde{Q}_1$ on the real and imaginary part of $\Lambda = diag(z_1, ..., z_m) = diag(\cos\theta_1, ..., \cos\theta_m) + i\ diag(\sin\theta_1, ..., \sin\theta_m)$ is considered. This motivated the following theorem [5, 7]:

THEOREM 2.1. *Let $n = 2\ell + 1 < m$. Let $C, S \in \mathbb{R}^{m\times m}$ be symmetric matrices with $C^2 + S^2 = I$ and $CS = SC$. Let $u = (\omega_1, ..., \omega_m)^T \in \mathbb{R}^m$ with $u^T u = 1$. There exists a unique $m \times n$ matrix $\hat{Q}$ with orthonormal columns such that*

$$\hat{Q}^T C\hat{Q} = X$$

ETNA
Kent State University
etna@mcs.kent.edu

$$
\begin{array}{rcl}
\hat{Q}^T S \hat{Q} & = & Y \\
\hat{Q} e_1 & = & u
\end{array}
$$

where X is a $(2\ell+1)\times(2\ell+1)$ symmetric pentadiagonal matrix of the form

$$
\left(\begin{array}{cccccccc}
x & x & \oplus & & & & & \\
x & x & x & \oplus & & & & \\
\oplus & x & x & x & \oplus & & & \\
 & \oplus & x & x & x & \oplus & & \\
 & & \ddots & \ddots & \ddots & \ddots & \ddots & \\
 & & & \oplus & x & x & x & \oplus \\
 & & & & \oplus & x & x & x \\
 & & & & & \oplus & x & x
\end{array}\right)
$$

with $x_{j+2,j} > 0$ and Y is a $(2\ell+1)\times(2\ell+1)$ symmetric bordered block tridiagonal matrix of the form

$$
\left(\begin{array}{c|cc|cc|cc|cc|cc}
x & \oplus & & & & & & & & & \\
\hline
\oplus & x & x & x & \ominus & & & & & & \\
 & x & x & x & x & & & & & & \\
\hline
 & x & x & x & x & x & \ominus & & & & \\
 & \ominus & x & x & x & x & x & & & & \\
\hline
 & & & \ddots & & \ddots & & \ddots & & & \\
\hline
 & & & & & x & x & x & x & x & \ominus \\
 & & & & & \ominus & x & x & x & x & x \\
\hline
 & & & & & & & x & x & x & x \\
 & & & & & & & \ominus & x & x & x
\end{array}\right)
\tag{2.3}
$$

with $y_{21} > 0, y_{2j+3,2j} < 0$ and $y_{2j+2,2j-1} = 0$. (Here x denotes any real-valued, $\oplus$ any positive and $\ominus$ any negative matrix element.)

The existence proof in [5] constructs the matrices $\widetilde{Q}_1, X$ and Y columnwise in a Lanczos-like style from the equations

$$
\begin{array}{rcl}
C\widetilde{q}_j & = & \widetilde{Q}_1 x_j \text{ and} \\
S\widetilde{q}_j & = & \widetilde{Q}_1 y_j.
\end{array}
$$

The first column of $\widetilde{Q}_1$ is given, the second column is found using the equation $S\widetilde{q}_1 = \widetilde{Q}_1 y_1$. The subsequent columns of $\widetilde{Q}_1$ can be computed using only the equation $C\widetilde{q}_j = \widetilde{Q}_1 x_j$. This construction leads to an $\mathrm{O}(mn)$ algorithm for the computation of $\widetilde{Q}_1, X$ and Y. A problem (as in every Lanczos-like algorithm) is the loss of orthogonality between the columns of $\widetilde{Q}_1$. In [5, 7] algorithms which require $\mathrm{O}(mn)$ arithmetic operations and avoid this problem are developed, as well as an algorithm to compute $\widetilde{R}_1^{-1}$ which requires $\mathrm{O}(n^2)$ operations. These algorithms construct the real-valued least-squares solution $\widetilde{t}$ of (2.2) in $\mathrm{O}(mn+n^2)$ arithmetic operations using only real arithmetic.

In [10] Newbery presents an algorithm for the discrete least-squares approximation by trigonometric functions which is closely related to the algorithms mentioned above. The algorithm solves the least-squares approximation problem (1.1) using real arithmetic. This $\mathrm{O}(n^2)$ algorithm is given in Section 3. Its connection to the algorithms

ETNA
Kent State University
etna@mcs.kent.edu

mentioned above is described in Section 4. It will be seen that Newbery's algorithm can be interpreted as constructing a QR decomposition of $D\widetilde{A}$ in (2.2). Especially the Q-factor of the QR decomposition is constructed columnwise in a Lanczos-like fashion similar to the way the matrix Q is constructed in the existence proof of Theorem 2.1.

3. The Newbery algorithm. In [10] Newbery develops an algorithm for computing least-squares approximation by trigonometric polynomials. Newbery constructs trigonometric polynomials $B_{00}, B_{10}, B_{11}, \ldots$, with

$$\begin{aligned} B_{00} &= 1, \\ B_{k,k-1} &= \sin k\phi + \sum_{j=0}^{k-1} (\widetilde{s}_j \sin j\phi + \widetilde{c}_j \cos j\phi), \text{ and} \\ B_{k,k} &= s_k \sin k\phi + \cos k\phi + \sum_{j=0}^{k-1} (\widehat{s}_j \sin j\phi + \widehat{c}_j \cos j\phi), \end{aligned} \tag{3.1}$$

orthogonal with respect to the inner product $(B_{rs}, B_{pq}) = \sum_{i=1}^{m} \omega_i^2 B_{rs}(\phi_i) B_{pq}(\phi_i)$ where the $\{\phi_j\}_{j=1}^{m}$ are, as before, m distinct values in the interval $[0, 2\pi)$.

Newbery presents a variant of the Gram-Schmidt-process to compute B_{00}, B_{10}, $B_{11}, \ldots$. Let

$$\begin{aligned} B &= [b_1, ..., b_{2l+1}] \\ &= \begin{pmatrix} B_{00}(\phi_1) & B_{10}(\phi_1) & B_{11}(\phi_1) & \cdots & B_{l,l-1}(\phi_1) & B_{ll}(\phi_1) \\ B_{00}(\phi_2) & B_{10}(\phi_2) & B_{11}(\phi_2) & \cdots & B_{l,l-1}(\phi_2) & B_{ll}(\phi_2) \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ B_{00}(\phi_m) & B_{10}(\phi_m) & B_{11}(\phi_m) & \cdots & B_{l,l-1}(\phi_m) & B_{ll}(\phi_m) \end{pmatrix}, \end{aligned} \tag{3.2}$$

$C = diag(\cos\phi_1, ..., \cos\phi_m)$, $S = diag(\sin\phi_1, ..., \sin\phi_m)$ and $b_1 = (1, ..., 1)^T$. Then the algorithm for constructing B_{00}, B_{10}, $B_{11}, \ldots$ can be summarized as follows (for a detailed discussion see [10]).

Newbery's algorithm

input : $\phi_1, ..., \phi_m, \omega_1, ..., \omega_m$
output : B as defined in (3.2)
    $s_0 = 0$
    for $k = 0, 1, 2, ..., l$
        if $k = 0$
        then $x = Cb_1$
            $y = Sb_1$
        else $x = 2Cb_{2k+1}$
            $y = 2Sb_{2k+1}$
        endif
        $\alpha_{k2} = (x, b_{2k-1})/(b_{2k-1}, b_{2k-1})$
        $\alpha_{k3} = (x, b_{2k})/(b_{2k}, b_{2k})$
        $\alpha_{k4} = (x, b_{2k+1})/(b_{2k+1}, b_{2k+1})$
        $\beta_{k1} = (y, b_{2k-2})/(b_{2k-2}, b_{2k-2})$
        $\beta_{k2} = (y, b_{2k-1})/(b_{2k-1}, b_{2k-1})$
        $\beta_{k3} = (y, b_{2k})/(b_{2k}, b_{2k})$
        $\beta_{k4} = (y, b_{2k+1})/(b_{2k+1}, b_{2k+1})$
        $x' = x - \alpha_{k2} b_{2k-1} - \alpha_{k3} b_{2k} - \alpha_{k4} b_{2k+1}$
        $y' = y - \beta_{k1} b_{2k-2} - \beta_{k2} b_{2k-1} - \beta_{k3} b_{2k} - \beta_{k4} b_{2k+1}$
        $b_{2k+2} = (y' + s_k x')/(1 + s_k^2)$
        $\alpha_{k5} = (x', b_{2k+2})/(b_{2k+2}, b_{2k+2})$
        $b_{2k+3} = x' - \alpha_{k5} b_{2k+2}$
        $s_{k+1} = s_k - \alpha_{k5}$

ETNA
Kent State University
etna@mcs.kent.edu

As Newbery states "Now that the orthogonal functions have been generated, the rest of the curve-fitting process will follow the conventional pattern" [10, p. 875,line 5-6].

4. Connection between the algorithms. First we will see that Newbery's algorithm can be interpreted as computing the QR decompositon of $D\widetilde{A}$. Using (3.1), $\cos k\phi$ and $\sin k\phi$ can be represented by

$$\begin{aligned}\cos k\phi &= \sum_{j=1}^{k}(\zeta_{kj}B_{jj}+\eta_{kj}B_{j,j-1})+\zeta_{k0}B_{00} \text{ and}\\ \sin k\phi &= B_{k,k-1}+\sum_{j=1}^{k-1}(\mu_{kj}B_{jj}+\nu_{kj}B_{j,j-1})+\mu_{k0}B_{00},\end{aligned}$$

with properly chosen $\zeta_{kj},\eta_{kj},\mu_{kj}$ and $\nu_{kj}\in\mathbb{R}$ ($\zeta_{kk}=1$). For $D\widetilde{A}$ from (2.2) we get $D\widetilde{A}=DBR$ where B as in (3.2) and

$$R=\begin{pmatrix} 1 & \mu_{10} & \zeta_{10} & \cdots & \mu_{l0} & \zeta_{l0}\\ 0 & 1 & \eta_{11} & \cdots & \nu_{l1} & \eta_{l1}\\ 0 & 0 & 1 & \cdots & \mu_{l1} & \zeta_{l1}\\ \vdots & \vdots & \vdots & & \vdots & \vdots\\ 0 & 0 & 0 & & 1 & \eta_{ll}\\ 0 & 0 & 0 & \cdots & 0 & 1 \end{pmatrix}.$$

Further we obtain

$$(DB)^H(DB)=diag((B_{00},B_{00}),(B_{10},B_{10}),(B_{11},B_{11}),...,(B_{ll},B_{ll})).$$

Let

$$W=diag(\sqrt{(B_{00},B_{00})},\sqrt{(B_{10},B_{10})},\sqrt{(B_{11},B_{11})},...,\sqrt{(B_{ll},B_{ll})}).$$

Then $WW=(DB)^H(DB)$ and

$$(DBW^{-1})^H(DBW^{-1})=I,$$

such that $D\widetilde{A}=(DBW^{-1})(WR)$ is a QR decomposition of $D\widetilde{A}$. As the diagonal elements of WR are positive we can obtain a unique "skinny" QR decomposition from $(DBW^{-1})(WR)$. Thus Newbery's algorithm implicitly computes the QR decomposition of $D\widetilde{A}$ in (2.2) and therefore solves the approximation problem (1.2).

Modifying Newbery's algorithm such that DBW^{-1} is computed instead of B, we can see that the algorithm can be regarded as a Lanczos-like algorithm. For this consider

$$CQ=QX,\qquad SQ=QY \tag{4.1}$$

columnwise where $C=diag(\cos\theta_1,...,\cos\theta_m)$, and $S=diag(\sin\theta_1,...,\sin\theta_m)$ as before, X is a symmetric pentadiagonal matrix with positive entries in the second subdiagonal, and Y is a symmetric bordered block tridiagonal matrix of the form

ETNA
Kent State University
etna@mcs.kent.edu

(2.3) as in Theorem 2.1. The odd numbered columns $1, 3, 5, ..., 2l+1$ of the above equations (4.1) give

$$\begin{aligned} Cq_{2k+1} &= Qx_{2k+1} \\ &= x_{2k-1,2k+1}q_{2k-1} + x_{2k,2k+1}q_{2k} + x_{2k+1,2k+1}q_{2k+1} \\ &\quad + x_{2k+2,2k+1}q_{2k+2} + x_{2k+3,2k+1}q_{2k+3} \text{ and} \end{aligned} \tag{4.2}$$

$$\begin{aligned} Sq_{2k+1} &= Qy_{2k+1} \\ &= y_{2k-2,2k+1}q_{2k-2} + y_{2k-1,2k+1}q_{2k-1} + y_{2k,2k+1}q_{2k} \\ &\quad + y_{2k+1,2k+1}q_{2k+1} + y_{2k+2,2k+1}q_{2k+2} + y_{2k+3,2k+1}q_{2k+3}. \end{aligned} \tag{4.3}$$

Assume that $q_1, q_2, ...q_{2k+1}$ are known, then

$$\begin{aligned} x_{j,2k+1} &= q_j^T C q_{2k+1}, \ j = 2k-1, 2k, 2k+1 \text{ and} \\ y_{j,2k+1} &= q_j^T S q_{2k+1}, \ j = 2k-2, 2k-1, 2k, 2k+1 \end{aligned}$$

can be computed. This corresponds to the computation of α_{k2}, α_{k3}, α_{k4}, β_{k1}, β_{k2}, β_{k3}, β_{k4} in the modified Newbery algorithm. The terms x' and y' of the modified Newbery algorithm correspond to the following transformations of (4.2), (4.3)

$$\begin{aligned} \breve{x} &:= Cq_{2k+1} - x_{2k-1,2k+1}q_{2k-1} - x_{2k,2k+1}q_{2k} - x_{2k+1,2k+1}q_{2k+1} \\ &= x_{2k+2,2k+1}q_{2k+2} + x_{2k+3,2k+1}q_{2k+3}, \\ \breve{y} &:= Sq_{2k+1} - y_{2k-2,2k+1}q_{2k-2} - y_{2k-1,2k+1}q_{2k-1} - y_{2k,2k+1}q_{2k} - y_{2k+1,2k+1}q_{2k+1} \\ &= y_{2k+2,2k+1}q_{2k+2} + y_{2k+3,2k+1}q_{2k+3}. \end{aligned}$$

The computation of b_{2k+2} and b_{2k+3} in the modified Newbery algorithm corresponds to the computation of q_{2k+2} and q_{2k+3} respectively. Using

$$\breve{x} - x_{2k+2,2k+1}q_{2k+2} = x_{2k+3,2k+1}q_{2k+3}$$

in the equation for $\breve{y}$ yields

$$\begin{aligned} \breve{y} - \frac{y_{2k+3,2k+1}}{x_{2k+3,2k+1}}\breve{x} &= (y_{2k+2,2k+1} - \frac{x_{2k+2,2k+1}}{x_{2k+3,2k+1}})q_{2k+2} \\ &= aq_{2k+2}, \end{aligned}$$

where

$$a := ||\breve{y} - \frac{y_{2k+3,2k+1}}{x_{2k+3,2k+1}}\breve{x}||.$$

So far we have computed neither $y_{2k+3,2k+1}$ nor $x_{2k+3,2k+1}$. Let us assume for the moment that $\frac{y_{2k+3,2k+1}}{x_{2k+3,2k+1}}$ can be determined. Thus we are assuming that q_{2k+2} is known. Then we can compute

$$x_{2k+2,2k+1} = q_{2k+2}^T C q_{2k+1}$$

(corresponds to α_{k5}), as well as

$$x_{2k+3,2k+1}q_{2k+3} = \breve{x} - x_{2k+2,2k+1}q_{2k+2}$$

where

$$x_{2k+3,2k+1} = ||\breve{x} - x_{2k+2,2k+1}q_{2k+2}||.$$

ETNA
Kent State University
etna@mcs.kent.edu

Thus we have shown that in case we can compute $\frac{y_{2k+3,2k+1}}{x_{2k+3,2k+1}}$, Newbery's algorithm can be interpreted as computing a matrix Q with orthonormal columns that simultaneously transforms the matrix $C = diag(\cos\theta_1, ..., \cos\theta_m)$ to symmetric pentadiagonal form and the matrix $S = diag(\sin\theta_1, ..., \sin\theta_m)$ to symmetric bordered block tridiagonal form (2.3).

It remains to be shown that we can compute $\frac{y_{2k+3,2k+1}}{x_{2k+3,2k+1}}$. As $q_1, q_2, ...q_{2k+1}$ are known, we have from (4.1) for the (2k-1)st columns

$$\begin{aligned} q_{2k+1} &= x_{2k+1,2k-1}^{-1}(Cq_{2k-1} + x_{2k-3,2k-1}q_{2k-3} + x_{2k-2,2k-1}q_{2k-2} \\ &\qquad\qquad + x_{2k-1,2k-1}q_{2k-1} + x_{2k,2k-1}q_{2k}), \\ Sq_{2k-1} &= \sum_{j=-2}^{3} y_{2k+j-2,2k-1}q_{2k+j-2}, \end{aligned}$$

and

$$\begin{aligned} x_{2k+j-2,2k+3} &= 0, \qquad\qquad j = -2, -1, 0, 1, 2, \\ x_{2k+3,2k+1} &\neq 0, \\ y_{2k+1,2k-1} &\neq 0. \end{aligned}$$

Thus we obtain

$$\begin{aligned} y_{2k+3,2k+1} &= q_{2k+3}^T S q_{2k+1} \\ &= x_{2k+1,2k-1}^{-1} q_{2k+3}^T CS q_{2k-1} \\ &= x_{2k+1,2k-1}^{-1} \sum_{j=-2}^{3} y_{2k+j-2,2k-1} q_{2k+3}^T C q_{2k+j-2} \\ &= x_{2k+1,2k-1}^{-1} \sum_{j=-2}^{3} y_{2k+j-2,2k-1} x_{2k+3,2k+j-2} \\ &= x_{2k+1,2k-1}^{-1} y_{2k+1,2k-1} x_{2k+3,2k+1}, \end{aligned}$$

such that

$$\frac{y_{2k+3,2k+1}}{x_{2k+3,2k+1}} = \frac{y_{2k+1,2k-1}}{x_{2k+1,2k-1}}.$$

Hence $\frac{y_{2k+3,2k+1}}{x_{2k+3,2k+1}}$ can be determined. Therefore we have shown that Newbery's algorithm can be regarded as a Lanczos-like algorithm. It builds up the matrix Q columnwise from the equations (4.1) using only the odd-numbered columns. Newbery's algorithm can be seen as computing implicitly the QR decomposition of $D\widetilde{A}$ (2.2).

5. Concluding Remarks. Numerical experiments that compare the different methods for solving the trigonometric approximation problem (1.2) are given in [5, 7]. These experiments have shown that generally the method proposed in [5, 7] which solves (2.2) using complex arithmetic via the inverse eigenproblem $Q(\Lambda - \lambda I)P = G_o - \lambda G_e$ produces more accurate results than the method proposed by Reichel, Ammar and Gragg (we will call this method RAG for the rest of this section). On the other hand, the method proposed in [5, 7] requires about 3 times as much time to solve the problem than the method RAG. Numerical tests showed that Newbery's method did not produce good results for all test examples. The method uses only the odd-numbered columns of C and S to build up the matrix Q columnwise from the equations (4.1), this theoretically introduces zeros in the even-numbered columns of X and Y. These theoretically generated zeros are affected with increasing rounding errors. The method proposed in [5, 7] which solves (2.2) using only real arithmetic circumvents this problem by using all columns of C and S to build up the matrix Q from (4.1). The accuracy of this method is about the same as the accuracy of the method RAG, but RAG is significantly faster.

REFERENCES

[1] G. S. Ammar, W. B. Gragg, and L. Reichel, *On the Eigenproblem for Orthogonal Matrices*, in Proc. 25th IEEE Conference on Decision and Control, 1986, pp. 1963-1966.

[2] ———, *Constructing a Unitary Hessenberg Matrix from Spectral Data*, in Numerical Linear Algebra, Digital Signal Processing, and Parallel Algorithms, G. Golub and P. VanDooren, eds., Springer-Verlag, Berlin, 1991, pp. 385-396.

[3] J.-P. Berrut, *Baryzentrische Formeln zur trigonometrischen Interpolation I + II*, Appl. Math. Phys. (ZAMP), 35 (1984), pp. 91 – 105, 193-205.

[4] Å. Björck and V. Pereyra, *Solution of Vandermonde systems of equations*, Math. Comp., 24 (1970), pp. 893-903.

[5] H. Fassbender, *Numerische Verfahren zur diskreten trigonometrischen Polynomapproximation*, Dissertation Universität Bremen, 1993.

[6] ———, *Inverse unitary eigenproblems and related orthogonal functions*, to appear in Numer. Math.

[7] ———, *On Numerical Methods For Discrete Least-Squares Approximation By Trigonometric Polynomials*, to appear in Math. Comp. 1997.

[8] G. H. Golub and C. F. van Loan, *Matrix Computation*, The John Hopkins University Press, second ed., 1989.

[9] P. Henrici, *Applied and Computational Analysis*, vol. 3, Wiley, 1986.

[10] A. C. R. Newbery, *Trigonometric Interpolation and Curve-Fitting*, Math. Comp., 24 (1970), pp. 869-876.

[11] L. Reichel, G. S. Ammar, and W. B. Gragg, *Discrete Least Squares Approximation by Trigonometric Polynomials*, Math. Comp., 57 (1991), pp. 273-289.