Electronic Transactions on Numerical Analysis.
Volume 3, pp. 96-115, September 1995.

ISSN 1068-9613.

ETNA
Kent State University
etna@mcs.kent.edu

GAUSSIAN QUADRATURE FOR MATRIX VALUED FUNCTIONS ON THE UNIT CIRCLE *

ANN SINAP†

Abstract. The Gaussian quadrature formulas for matrix valued functions on the unit circle are described. It is shown how the eigenvalues and eigenvectors of a unitary lower block Hessenberg matrix can be used to compute an approximation of a given matrix integral on the unit circle. A parallel algorithm for this purpose has been implemented on a IBM SP1 and some examples are worked out.

Key words. orthogonal matrix polynomials, block Hessenberg matrices, quadrature, parallel algorithm.

AMS subject classifications. 42C05, 41A55, 47A56, 65D32, 65Y05.

1. Introduction.

The aim of this paper is to discuss the computation of Gaussian quadrature rules for matrix-valued functions on the unit circle on a high performance computer. These quadrature formulas generalize the scalar formulas considered in [12]. In [1], [10] and [11] the authors describe a divide and conquer method for the solution of the unitary Hessenberg eigenproblem, which is essentially the problem to be solved here. We will describe a divide and conquer method for the unitary block Hessenberg problem which is applied to quadrature rules for matrix-valued functions. The algorithm is implemented on an IBM SP1 multiprocessor which consists of a few connected RISC-processors, each of which has its own memory and disk. This gives a multiprocessor with distributed memory. In order to facilitate communication between different processors, the user needs instructions which he can call from his C or Fortran programs. These instructions are part of a message passing library. We use PVM, Parallel Virtual Machine, which was developed at the Oak Ridge National Laboratory and is publicly available.

In Section 2 we introduce orthogonal matrix polynomials on the unit circle and discuss some of their properties. Orthogonal matrix polynomials previously have been considered by Delsarte, Genin and Kamp [5] and Geronimo [6]. In Section 3 we describe the Gaussian quadrature rules and explain how the eigenvalues and eigenvectors of a unitary lower block Hessenberg matrix can be used to determine the quadrature weights. In Section 4 we describe an implementation of the quadrature formulas on a distributed memory multiprocessor, using a divide and conquer method to compute the spectral factorization of a unitary lower block Hessenberg matrix.

2. Orthogonal matrix polynomials on the unit circle.

Let ρ be a matrix-valued distribution function on $[0, 2\pi)$, which defines a matrix-valued measure on the unit circle (see [5], [6]). We introduce two inner products: a *left* inner product and a *right* inner product. The left inner product is given by

$$\langle P, Q\rangle_L := \frac{1}{2\pi}\int_0^{2\pi} P(z)\, d\rho(\theta)\, Q(z)^*, \qquad z = e^{i\theta},$$

* Received April 25, 1995. Accepted for publication September 12, 1995. Communicated by L. Reichel.
† Department of Mathematics, Katholieke Universiteit Leuven, Celestijnenlaan 200 B, B–3001 Heverlee, BELGIUM, (Ann.Sinap@wis.kuleuven.ac.be).

where $P, Q \in \mathbb{C}^{p\times p}[z]$, the set of $p \times p$ matrix polynomials in a complex variable z and A^* denotes the Hermitian conjugate of the matrix A. The right inner product is given by

$$\langle P, Q\rangle_R \;:=\; \frac{1}{2\pi}\int_0^{2\pi} P(z)^*\, d\rho(\theta)\, Q(z), \qquad z = e^{i\theta}.$$

Now we are able to define orthogonal matrix polynomials. Left orthonormal matrix polynomials ϕ_n^L, $n = 0, 1, 2, \ldots$, are obtained by orthonormalizing $I, zI, z^2 I, \ldots$ with respect to the left inner product,

$$\frac{1}{2\pi}\int_0^{2\pi} \phi_n^L(z)\, d\rho(\theta)\, \phi_m^L(z)^* \;=\; \delta_{n,m}\, I, \qquad n, m \geq 0.$$

Similarly right orthonormal matrix polynomials ϕ_n^R, $n = 0, 1, 2, \ldots$, are obtained using the right inner product,

$$\frac{1}{2\pi}\int_0^{2\pi} \phi_n^R(z)^*\, d\rho(\theta)\, \phi_m^R(z) \;=\; \delta_{n,m}\, I, \qquad n, m \geq 0.$$

Both ϕ_n^L and ϕ_n^R are matrix polynomials of degree n, with a non-singular leading coefficient. The left orthonormal matrix polynomials are unique up to a multiplication on the left by a unitary matrix, while the right orthonormal matrix polynomials are defined up to a multiplication on the right by a unitary matrix. Since there is no simple relation between left and right orthonormal polynomials we need to analyse both.

Without loss of generality we only consider matrix distribution functions that satisfy

$$\frac{1}{2\pi}\int_0^{2\pi} d\rho(\theta) \;=\; I.$$

In order to get the *reversed polynomial* $\tilde{P}(z)$, we take the polynomial P, reverse the order of the coefficients and take the Hermitian conjugates of these coefficients. That is,

$$\tilde{P}(z) := z^n P\left(\frac{1}{\bar{z}}\right)^*.$$

These orthonormal matrix polynomials satisfy some simple recurrence relations (see [5], [6], [15]):

$$(I - H_n H_n^*)^{\frac{1}{2}} \phi_n^L(z) = z\phi_{n-1}^L(z) + H_n \tilde{\phi}_{n-1}^R(z),$$

and

$$(I - H_n^* H_n)^{\frac{1}{2}} \tilde{\phi}_n^R(z) = \tilde{\phi}_{n-1}^R(z) + zH_n^* \phi_{n-1}^L(z),$$

where $(I - H_n H_n^*)^{\frac{1}{2}}$ and $(I - H_n^* H_n)^{\frac{1}{2}}$ are positive definite. We can show the zeros of these polynomials are located in the open unit disk, and some straightforward matrix computations give us the identity of Christoffel-Darboux for orthonormal matrix polynomials on the unit circle:

THEOREM 2.1.

$$(1-\bar{z}_1 z_2)\sum_{i=0}^{n}\phi_i^L(z_1)^*\,\phi_i^L(z_2) = \tilde{\phi}_{n+1}^R(z_1)^*\,\tilde{\phi}_{n+1}^R(z_2)-\phi_{n+1}^L(z_1)^*\,\phi_{n+1}^L(z_2),$$
$$= \tilde{\phi}_n^R(z_1)^*\,\tilde{\phi}_n^R(z_2)-\bar{z}_1 z_2\,\phi_n^L(z_1)^*\,\phi_n^L(z_2),$$

and

$$(1-z_1\bar{z}_2)\sum_{i=0}^{n}\phi_i^R(z_1)\,\phi_i^R(z_2)^* = \tilde{\phi}_{n+1}^L(z_1)\,\tilde{\phi}_{n+1}^L(z_2)^*-\phi_{n+1}^R(z_1)\,\phi_{n+1}^R(z_2)^*$$
$$= \tilde{\phi}_n^L(z_1)\,\tilde{\phi}_n^L(z_2)^*-z_1\bar{z}_2\,\phi_n^R(z_1)\,\phi_n^R(z_2)^*.$$

3. Gaussian quadrature on the unit circle. The integral of an arbitrary matrix function can be approximated by means of a sum of the form

$$\frac{1}{2\pi}\int_0^{2\pi}F(z)\,d\rho(\theta)\,G(z)^*\simeq\sum_{i=1}^{k}F(z_i)\,\Lambda_i G(z_i)^*,$$

where $\Lambda_i\in\mathbb{C}^{p\times p}$. It will be convenient to choose

$$\Lambda_i=\begin{pmatrix} v_{i,1} & \dots & v_{i,m_i}\end{pmatrix} A_i \begin{pmatrix} v_{i,1}^* \\ \vdots \\ v_{i,m_i}^*\end{pmatrix},$$

where the non-zero vectors $v_{i,1},\ldots,v_{i,m_i}$ are linearly independent, the points z_i lie on the unit circle and $\sum_{i=1}^k m_i=np$. We would like to have a formula which is exact for as many Laurent polynomials as possible. Denote by $\Lambda_{-m,n}$ the set of Laurent polynomials of the form $\sum_{i=-m}^{n}A_i z^i$. We restrict ourselves to the left inner product $\langle .,.\rangle_L$, but the procedure can be repeated for the right inner product, $\langle .,.\rangle_R$, as well.

Since the zeros of the orthonormal matrix polynomials lie in the open unit disk, we cannot use these points in the quadrature formula. We therefore introduce para-orthogonal matrix polynomials

$$B_n(z,W_n):=\phi_n^L(z)+W_n\tilde{\phi}_n^R(z),$$

where W_n is a unitary matrix.

THEOREM 3.1. *The zeros of the para-orthogonal matrix polynomials $B_n(\cdot,W_n)$ are the eigenvalues of a unitary lower block Hessenberg matrix.*

Proof. Expand the left orthonormal matrix polynomial ϕ_n^L into a Fourier series using the left orthonormal matrix polynomials ϕ_i^L, $i=0,1,\ldots,n-1$ and the para-orthogonal matrix polynomial $B_n(\cdot,W_n)$. This yields,

$$z\,\phi_i^L(z) = \sum_{j=0}^{i+1}M_{i,j}\phi_j^L(z),\qquad i=0,1,\ldots,n-2,\ \text{ and}$$
$$z\,\phi_{n-1}^L(z) = \sum_{j=0}^{n-1}M_{n-1,j}\phi_j^L(z)+M_{n-1,n}B_n(z,W_n).$$

ETNA
Kent State University
etna@mcs.kent.edu

If z_0 is a zero of $B_n(\cdot, W_n)$ with root vector v_0, then

$$\begin{aligned} z_0\, \phi_i^L(z_0)v_0 &= \sum_{j=0}^{i+1} M_{i,j}\phi_j^L(z_0)v_0, \qquad i = 0,1,\ldots,n-2, \quad \text{and} \\ z_0\, \phi_{n-1}^L(z_0)v_0 &= \sum_{j=0}^{n-1} M_{n-1,j}\phi_j^L(z_0)v_0, \end{aligned}$$

and hence z_0 is an eigenvalue of the block Hessenberg matrix

$$M_n := \begin{pmatrix} M_{0,0} & M_{0,1} & & & & \\ M_{1,0} & M_{1,1} & M_{1,2} & & & \\ \vdots & \vdots & \ddots & \ddots & & \\ \vdots & \vdots & \vdots & \ddots & \ddots & \\ M_{n-2,0} & M_{n-2,1} & \ldots & \ldots & M_{n-2,n-2} & M_{n-2,n-1} \\ M_{n-1,0} & M_{n-1,1} & \ldots & \ldots & M_{n-1,n-2} & M_{n-1,n-1} \end{pmatrix}.$$

The blocks of this matrix satisfy

$$\begin{aligned} M_{i,j} &= \langle z\phi_i^L, \phi_j^L\rangle_L, \quad i = 0,1,\ldots,n-2, \quad j = 0,1,\ldots,i+1, \quad \text{and} \\ M_{n-1,j} &= \langle z\phi_{n-1}^L, \phi_j^L\rangle_L - M_{n-1,n}W_n\langle \tilde{\phi}_n^R, \phi_j^L\rangle_L, \quad j = 0,1,\ldots,n-1, \quad \text{and} \\ M_{n-1,n} &= \langle z\phi_{n-1}^L, \phi_n^L\rangle_L (I + W_nH_n^*)^{-1}. \end{aligned}$$

From the properties of the left inner product and orthogonality we get

$$\langle z\phi_k^L, z\phi_l^L\rangle_L = \langle \phi_k^L, \phi_l^L\rangle_L = \delta_{k,l}I, \qquad 0 \le k,l \le n-1,$$

so that for $0 \le k,l \le n-2$,

$$\sum_{j=0}^{\min(k,l)+1} M_{k,j}M_{l,j}^* = \langle z\phi_k^L, z\phi_l^L\rangle_L = \delta_{k,l}I.$$

Since $\langle B_n(\cdot,W_n), z\phi_k^L\rangle_L = 0$, $k = 0,1,\ldots,n-2$, we get

$$\sum_{j=0}^{k+1} M_{n-1,j}M_{k,j}^* = \langle z\phi_{n-1}^L, z\phi_k^L\rangle_L = 0.$$

Finally,

$$\sum_{k=0}^{n-1} \langle B_n(\cdot,W_n), \phi_k^L\rangle_L M_{n-1,k}^* = (I + H_nW_n^*)\, M_{n-1,n}^* - \langle B_n(\cdot,W_n), B_n(\cdot,W_n)\rangle_L M_{n-1,n}^*$$

and the equation

$$\langle B_n(\cdot,W_n), B_n(\cdot,W_n)\rangle_L = 2I + W_nH_n^* + H_nW_n^*$$

show us that

$$\sum_{l=0}^{n-1} M_{n-1,l}M_{n-1,l}^* = \langle z\phi_{n-1}^L, z\phi_{n-1}^L\rangle_L = I. \qquad \square$$

ETNA
Kent State University
etna@mcs.kent.edu

Thus, the zeros of the para-orthogonal matrix polynomials lie on the unit circle. The following theorem shows a further related property.

THEOREM 3.2. *The multiplicity of the zeros of $B_n(\cdot, W_n)$ is less than or equal to p and the length of the corresponding Jordan chains is exactly* 1.

Proof. Let z_0 be a zero of $B_n(\cdot, W_n)$ with multiplicity $> p$, and let $\{(v_{i,0}, v_{i,1}, \ldots, v_{i,\mu_i-1}) : i = 1, 2, \ldots, s\}$ be a canonical set of right Jordan chains corresponding to this zero z_0. Since the multiplicity is greater than p, there has to be a least one Jordan chain with length > 1. Suppose $v_{i,0}, v_{i,1}$ are the leading vectors of this Jordan chain. Then

$$\begin{aligned} &B_n(z_0, W_n)v_{i,0} = 0, \qquad v_{i,0} \neq 0, \quad \text{and} \\ &B_n'(z_0, W_n)v_{i,0} + B_n(z_0, W_n)v_{i,1} = 0. \end{aligned}$$

The first relation shows that

$$\phi_n^L(z_0)v_{i,0} = -W_n\tilde{\phi}_n^R(z_0)v_{i,0},$$

and from the identity of Christoffel-Darboux we get

$$\sum_{i=0}^{n-1} \phi_i^L(z_1)^* \, \phi_i^L(z_1) = z_1 \left(\phi_n^L(z_1)^* \, \phi_n^L(z_1)' - \tilde{\phi}_n^R(z_1)^* \, \tilde{\phi}_n^R(z_1)' \right),$$

where $|z_1| = 1$. We obtain

$$\begin{aligned} v_{i,0}^* \sum_{i=0}^{n-1} \phi_i^L(z_0)^* \, \phi_i^L(z_0) &= -z_0 v_{i,0}^* \tilde{\phi}_n^R(z_0)^* W_n^* B_n'(z_0, W_n) \\ &= z_0 v_{i,0}^* \phi_n^L(z_0)^* B_n'(z_0, W_n). \end{aligned}$$

On the other hand the identity of Christoffel-Darboux gives us

$$0 = \tilde{\phi}_n^R(z_0)^* \, \tilde{\phi}_n^R(z_0) - \phi_n^L(z_0)^* \, \phi_n^L(z_0),$$

or

$$\begin{aligned} 0 &= v_{i,0}^* \tilde{\phi}_n^R(z_0)^* \left(\tilde{\phi}_n^R(z_0) + W_n^* \phi_n^L(z_0) \right) \\ &= v_{i,0}^* \tilde{\phi}_n^R(z_0)^* W_n^* B_n(z_0, W_n) \\ &= -v_{i,0}^* \phi_n^L(z_0)^* B_n(z_0, W_n). \end{aligned}$$

But then we have

$$\begin{aligned} v_{i,0}^* \sum_{i=0}^{n-1} \phi_i^L(z_0)^* \, \phi_i^L(z_0) v_{i,0} &= z_0 v_{i,0}^* \phi_n^L(z_0)^* B_n'(z_0, W_n) v_{i,0} \\ &= -z_0 v_{i,0}^* \phi_n^L(z_0)^* B_n(z_0, W_n) v_{i,1} \\ &= 0, \end{aligned}$$

and since $v_{i,0} \neq 0$ and $\sum_{i=0}^{n-1} \phi_i^L(z_0)^* \, \phi_i^L(z_0)$ is positive definite, this is impossible. ☐

The theorem below shows that the Gaussian quadrature rules, for the matrix case, are analogous to the rules for the scalar case (see [14]).

THEOREM 3.3. *Let (X, J) be a Jordan pair of the para-orthogonal matrix polynomial $B_n(z, W_n) = \phi_n^L(z) + W_n \tilde{\phi}_n^R(z)$, where $\{\phi_n^L\}$ is the set of the orthonormal matrix polynomials with respect to the matrix inner product $\langle ., . \rangle_L$, $\{\phi_n^R\}$ is the set with respect to the matrix inner product $\langle ., . \rangle_R$, and W_n is a unitary matrix. Then we have*

$$\langle F, G \rangle_L = \frac{1}{2\pi} \int_0^{2\pi} F(z)\, d\rho(\theta)\, G(z)^* \simeq \sum_{i=1}^{k} F(z_i)\, \Lambda_i\, G(z_i)^*, \qquad z = e^{i\theta},$$

where k is the number of distinct zeros z_i of $B_n(\cdot, W_n)$, m_i is the multiplicity of z_i, $v_{i,j}$ are the vectors associated with z_i,

$$\Lambda_i = \begin{pmatrix} v_{i,1} & \dots & v_{i,m_i} \end{pmatrix} {K_i}^{-1} \begin{pmatrix} v_{i,1}^* \\ \vdots \\ v_{i,m_i}^* \end{pmatrix},$$

$$K_i = \begin{pmatrix} v_{i,1}^* \\ \vdots \\ v_{i,m_i}^* \end{pmatrix} S_{n-1}^L(z_i, z_i) \begin{pmatrix} v_{i,1} & \dots & v_{i,m_i} \end{pmatrix},$$

and

$$S_{n-1}^L(z_1, z_2) = \sum_{j=0}^{n-1} \phi_j^L(z_2)^* \phi_j^L(z_1).$$

This quadrature formula is exact for Laurent matrix polynomials

$$F \in \Lambda_{-s,t} \qquad \textit{and} \qquad G \in \Lambda_{-(n-1-t),(n-1-s)},$$

where $0 \leq s, t \leq n - 1$.

In order to compute the quadrature weights we need to know the zeros and root vectors of the para-orthogonal matrix polynomials, but it has been shown that the zeros of $B_n(\cdot, W_n)$ correspond to the eigenvalues of a lower block Hessenberg matrix M_n. This property leads to the following theorem, which can be established similarly to the analogous result for the scalar case considered in [13].

THEOREM 3.4. *Let $U^{(i,j)}$, $j = 1, 2, \dots, m_i$, be the eigenvectors of the matrix M_n associated with the eigenvalue z_i. Then the associated quadrature coefficient is given by*

$$\Lambda_i = \begin{pmatrix} U_0^{(i,1)} & U_0^{(i,2)} & \dots & U_0^{(i,m_i)} \end{pmatrix} {G_i}^{-1} \begin{pmatrix} {U_0^{(i,1)}}^* \\ {U_0^{(i,2)}}^* \\ \vdots \\ {U_0^{(i,m_i)}}^* \end{pmatrix},$$

where

$$(G_i)_{s,t} = {U^{(i,s)}}^* U^{(i,t)},$$

and $U_0^{(i,j)}$ is the vector consisting of the first p components of $U^{(i,j)}$.

If $V^{(i,j)}$ are the normalized eigenvectors, then the quadrature coefficient is given by

$$\Lambda_i = \begin{pmatrix} V_0^{(i,1)} & V_0^{(i,2)} & \dots & V_0^{(i,m_i)} \end{pmatrix} \begin{pmatrix} {V_0^{(i,1)}}^T \\ {V_0^{(i,2)}}^T \\ \vdots \\ {V_0^{(i,m_i)}}^T \end{pmatrix},$$

where $V_0^{(i,j)}$ is the vector consisting of the first p components of $V^{(i,j)}$.

4. Algorithms to approximate a matrix integral. In view of the results of Section 3, Gaussian quadrature weights can be computed using the eigenvalues and the first p components of the orthonormalized eigenvectors of a unitary lower block Hessenberg matrix. The construction of a parallel algorithm for the approximation of a matrix integral on the unit circle will be primarily be concerned with the construction of a parallel algorithm for the computation of the eigensystem of a unitary lower block Hessenberg matrix. In [1], [10], [11] the problem was treated for the scalar case, $p = 1$. In those papers, the authors constructed a divide and conquer algorithm to solve the problem. In [3] a divide and conquer method for the eigenproblem of Hermitian matrices is described, and a transformation of a unitary matrix to a Hermitian matrix is given. We will combine these ideas and construct a divide and conquer algorithm for the block case.

4.1. A Divide and Conquer Method. In order to divide the problem into two smaller, but similar problems, we need some properties of M_n. First we show the unitary lower block Hessenberg matrix M_n can be written as a product of block Givens reflectors.

THEOREM 4.1. *The unitary lower block Hessenberg matrix M_n satisfies*

$$M_n = G_n'' G_{n-1} \cdots G_2 G_1,$$

where

$$G_i = \begin{pmatrix} I_{(i-1)p} & & & \\ & -H_i & (I - H_i H_i^*)^{1/2} & \\ & (I - H_i^* H_i)^{1/2} & H_i^* & \\ & & & I_{(n-i-1)p} \end{pmatrix}, \qquad 1 \le i \le n-1,$$

is unitary,

$$G_n'' = \begin{pmatrix} I_{(n-1)p} & \\ & K \end{pmatrix},$$

and $K = -\left((I - H_n H_n^*)^{-1/2} H_n + (I - H_n H_n^*)^{1/2} (W_n^* + H_n^*)^{-1} \right) (I - H_n^* H_n)^{1/2}$.

Proof. Some straightforward matrix computations show that the matrices G_i, $i = 1, 2, \dots, n-1$, are unitary and the blocks of M_n satisfy

$$\begin{aligned} M_{i,i+1} &= (I - H_{i+1} H_{i+1}^*)^{\frac{1}{2}}, \qquad 0 \le i \le n-2, \\ M_{i,j} &= -H_{i+1}(I - H_i^* H_i)^{\frac{1}{2}} (I - H_{i-1}^* H_{i-1})^{\frac{1}{2}} \dots (I - H_{j+1}^* H_{j+1})^{\frac{1}{2}} H_j^*, \\ &\qquad\qquad 0 \le i \le n-2,\ 0 \le j \le i, \\ M_{n-1,j} &= K\,(I - H_{n-1}^* H_{n-1})^{\frac{1}{2}} \dots (I - H_{j+1}^* H_{j+1})^{\frac{1}{2}} H_j^*, \qquad 0 \le j \le n-1. \end{aligned}$$

ETNA
Kent State University
etna@mcs.kent.edu

Multiplying M_n on the right with G_1^* and using the fact that M_n is unitary, gives

$$M_nG_1^* = \begin{pmatrix} I & 0 & 0 & \cdots & \cdots & 0 \\ 0 & -H_2 & (I-H_2H_2^*)^{1/2} & 0 & \cdots & 0 \\ 0 & M'_{2,1} & M_{2,2} & M_{2,3} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & M'_{n-1,1} & M_{n-1,2} & \cdots & \cdots & M_{n-1,n-1} \end{pmatrix},$$

where

$$\begin{aligned} M'_{j,1} &= M_{j,0}(I-H_1^*H_1)^{1/2} + M_{j,1}H_1 \\ &= -H_{j+1}(I-H_j^*H_j)^{1/2}\cdots(I-H_2^*H_2)^{1/2}, \quad 2 \le j \le n-2, \end{aligned}$$

and where

$$\begin{aligned} M'_{n-1,1} &= M_{n-1,0}(I-H_1^*H_1)^{1/2} + M_{n-1,1}H_1 \\ &= K(I-H_{n-1}^*H_{n-1})^{1/2}\cdots(I-H_2^*H_2)^{1/2}. \end{aligned}$$

Repeating this procedure $n-1$ times gives

$$M_nG_1^*\cdots G_{n-1}^* = \begin{pmatrix} I & & & & \\ & \ddots & & & \\ & & I & & \\ & & & I & \\ & & & & M'_{n-1,n-1} \end{pmatrix},$$

with

$$M'_{n-1,n-1} = M'_{n-1,n-2}(I-H_{n-1}^*H_{n-1})^{1/2} + M_{n-1,n-1}H_{n-1} = K.$$

So we get $M_nG_1^*\cdots G_{n-1}^* = G''_n$ or $M_n = G''_nG_{n-1}\cdots G_1$. □

The matrices G_i, $i = 1, 2, \ldots, n-1$, can be factored into three matrices, the middle of which is Hermitian.

THEOREM 4.2.

$$G_s = \begin{pmatrix} I_{(s-1)p} & & & \\ & I & & \\ & & R^* & \\ & & & I_{(n-s-1)p} \end{pmatrix} G'_s \begin{pmatrix} I_{(s-1)p} & & & \\ & R & & \\ & & I & \\ & & & I_{(n-s-1)p} \end{pmatrix},$$

where

$$G'_s = \begin{pmatrix} I_{(s-1)p} & & & \\ & -(H_sH_s^*)^{1/2} & (I-H_sH_s^*)^{1/2} & \\ & (I-H_sH_s^*)^{1/2} & (H_sH_s^*)^{1/2} & \\ & & & I_{(n-s-1)p} \end{pmatrix}.$$

The unitary matrix R is given by UV^*, where $H_s = U\Sigma V^*$ is the singular value decomposition of H_s.

Proof. The result follow from the properties of the singular value decomposition of H_s. □

The unitary Hermitian matrix G'_s can be written as a block Householder transformation.

THEOREM 4.3.

$$G'_s = I - WW^*$$

where

$$W = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ \left(I + (H_s H_s^*)^{1/2}\right)^{1/2} \\ -\left(I - (H_s H_s^*)^{1/2}\right)^{1/2} \\ 0 \\ \vdots \\ 0 \end{pmatrix} \in \mathbb{C}^{np\times p}.$$

Proof. The result can be established by straightforward computations. □

The properties shown allow us to divide the given problem into two smaller problems. If we define

$$M(H_1, H_2, \ldots, H_{n-1}, K) := G''_n G_{n-1} \cdots G_{s+1} G_s G_{s-1} \cdots G_1,$$

then we have

$$M_n = \begin{pmatrix} I_{sp} & \\ & M_n^{(2)} \end{pmatrix} G'_s \begin{pmatrix} M_n^{(1)} & \\ & I_{(n-s)p} \end{pmatrix},$$

where $M_n^{(1)}$ is a $sp \times sp$ matrix given by

$$M_n^{(1)} = M(H_1, H_2, \ldots, H_{s-1}, R),$$

and the $(n-s)p \times (n-s)p$ matrix $M_n^{(2)}$ satisfies

$$M_n^{(2)} = M(H_{s+1}R^*, H_{s+2}R^*, \ldots, H_{n-1}R^*, KR^*).$$

So we obtain

$$\begin{pmatrix} I_{sp} & \\ & M_n^{(2)*} \end{pmatrix} M_n \begin{pmatrix} I_{sp} & \\ & M_n^{(2)} \end{pmatrix} = G'_s \begin{pmatrix} M_n^{(1)} & \\ & M_n^{(2)} \end{pmatrix} = G'_s \tilde{M}_n.$$

This implies that the eigenvalues of M_n and $G'_s\tilde{M}_n$ are exactly the same, and if v is an eigenvector of $G'_s\tilde{M}_n$, then $\begin{pmatrix} I_{sp} & \\ & M_n^{(2)} \end{pmatrix} v$ is an eigenvector of M_n.

Hence, the computational problem is reduced to the following: given the eigensystems $M_n^{(i)} = \tilde{Q}_i \tilde{\Lambda}_i \tilde{Q}_i^*$, $i = 1, 2$, determine the spectral factorization of $G'_s\tilde{M}_n =$

$(I - WW^*)\tilde{M}_n$.

Suppose $1 \notin \sigma(\tilde{M}_n)$ and $1 \notin \sigma(M_n)$. If 1 belongs to one of the spectra, we rotate the spectra (see below) such that these conditions are fulfilled. Define

$$F := i(I - G'_s\tilde{M}_n)^{-1}(I + G'_s\tilde{M}_n). \tag{4.1}$$

This matrix is Hermitian, has the same eigenvectors as $G'_s\tilde{M}_n$ and its eigenvalues are the images of the eigenvalues of $G'_s\tilde{M}_n$ under the mapping

$$f(e^{i\theta}) := i\frac{1 + e^{i\theta}}{1 - e^{i\theta}} = \frac{-\sin\theta}{1 - \cos\theta} = -cotg\left(\frac{\theta}{2}\right),$$

which maps points of the unit circle to the real line.

Similarly we define the Hermitian matrix

$$\tilde{F} := i(I - \tilde{M}_n)^{-1}(I + \tilde{M}_n). \tag{4.2}$$

By means of computations similar to those in [3], we can show the following result.

THEOREM 4.4. *Let F and $\tilde{F}$ be defined by (4.1) and (4.2) respectively. Then they satisfy*

$$F - \tilde{F} = VDV^*,$$

where

$$V = (I - \tilde{M}_n)^{-1}W = \tilde{Q}(I - \tilde{\Lambda})^{-1}U,$$

and where

$$D^{-1} = U^*\begin{pmatrix} \ddots & & \\ & \frac{1}{4}cotg\left(\frac{\tilde{\theta}_j}{2}\right) & \\ & & \ddots \end{pmatrix}U,$$

and $U = \tilde{Q}^*W$.

Define $N_F(\lambda)$ as the number of eigenvalues of $F < \lambda$ and $N_{\tilde{F}}(\lambda)$ as the number of eigenvalues of $\tilde{F} < \lambda$. Let $pos(A)$ be the number of positive eigenvalues of A, $neg(A)$ the number of negative eigenvalues of A and $nul(A)$ the number of eigenvalues of A equal to 0. Let the matrix A^+ denote the Moore-Penrose pseudo-inverse of A.

THEOREM 4.5. *Let $F, \tilde{F} \in \mathbb{C}^{np\times np}$ be Hermitian matrices satisfying $F = \tilde{F} + VDV^*$, where $V \in \mathbb{C}^{np\times p}$ and $D \in \mathbb{C}^{p\times p}$ is non-singular.*

(i) *Let $\lambda \notin \sigma(\tilde{F})$. Then the matrices*

$$\begin{pmatrix} F - \lambda I & 0 \\ 0 & -D^{-1} \end{pmatrix} \quad and \quad \begin{pmatrix} \tilde{F} - \lambda I & 0 \\ 0 & -\left(D^{-1} + V^*(\tilde{F} - \lambda I)^{-1}V\right) \end{pmatrix}$$

are congruent and

$$N_F(\lambda) = N_{\tilde{F}}(\lambda) + pos(Z(\lambda)) - pos(D^{-1}),$$

where $Z(\lambda) = D^{-1} + V^*(\tilde{F} - \lambda I)^{-1}V$.

(ii) *Let $\lambda \in \sigma(\tilde{F})$ with multiplicity μ and let P be an orthonormal base of the corresponding eigenspace. Then the matrices*

$$\begin{pmatrix} F-\lambda I & 0 & 0 \\ 0 & -D^{-1} & 0 \\ 0 & 0 & 0_\mu \end{pmatrix} \text{ and } \begin{pmatrix} \tilde{F}-\lambda I & 0 & 0 \\ 0 & -\left(D^{-1}+V^*(\tilde{F}-\lambda I)^+ V\right) & -V^*P \\ 0 & -P^*V & 0 \end{pmatrix}$$

are congruent and

$$N_F(\lambda) = N_{\tilde{F}}(\lambda) + pos(Ze(\lambda)) - pos(D^{-1}),$$

where

$$Ze(\lambda) = \begin{pmatrix} D^{-1}+V^*(\tilde{F}-\lambda I)^+ V & V^*P \\ P^*V & 0 \end{pmatrix}.$$

Suppose we take all the arguments in the interval $[0, 2\pi)$. Then the function $f(\theta) = -cotg\left(\frac{\theta}{2}\right)$ is strictly increasing. If $N_{M_n}(\theta)$ is the number of eigenvalues of M_n with argument $< \theta$ and $N_{\tilde{M}_n}(\theta)$ is the number of eigenvalues of $\tilde{M}_n$ with argument $< \theta$, then

$$N_{M_n}(\theta) = N_F\left(-cotg\left(\frac{\theta}{2}\right)\right) \qquad N_{\tilde{M}_n}(\theta) = N_{\tilde{F}}\left(-cotg\left(\frac{\theta}{2}\right)\right).$$

These equations and the previous theorem imply that we can localize the eigenvalues of $G'_s\tilde{M}_n = (I - WW^*)\tilde{M}_n$.

THEOREM 4.6.

(i) *If $e^{i\theta} \notin \sigma(\tilde{M}_n)$, then*

$$N_{M_n}(\theta) = N_{\tilde{M}_n}(\theta) + pos\left(Z\left(-cotg\left(\frac{\theta}{2}\right)\right)\right) - pos(D^{-1}).$$

(ii) *If $e^{i\theta} \in \sigma(\tilde{M}_n)$ with multiplicity μ and if the corresponding orthonormal basis id P, then*

$$N_{M_n}(\theta) = N_{\tilde{M}_n}(\theta) + pos\left(Ze\left(-cotg\left(\frac{\theta}{2}\right)\right)\right) - pos(D^{-1}).$$

The matrices $Z\left(-cotg\left(\frac{\theta}{2}\right)\right)$ and $Ze\left(-cotg\left(\frac{\theta}{2}\right)\right)$ can be rewritten in terms of the spectral decompositions of the smaller matrices:

$$\begin{aligned} Z\left(-cotg\left(\frac{\theta}{2}\right)\right) &= D^{-1} + V^*\left(\tilde{F} + cotg\left(\frac{\theta}{2}\right) I\right)^{-1} V \\ &= D^{-1} + U^*(I-\tilde{\Lambda}^*)^{-1}\left(\tilde{\Lambda}_F + cotg\left(\frac{\theta}{2}\right) I\right)^{-1}(I-\tilde{\Lambda})^{-1}U \\ &= D^{-1} + R^* \begin{pmatrix} \ddots & & \\ & cotg\left(\frac{\theta}{2}\right) - cotg\left(\frac{\tilde{\theta}_j}{2}\right) & \\ & & \ddots \end{pmatrix}^{-1} R, \end{aligned}$$

ETNA
Kent State University
etna@mcs.kent.edu

where $R = \left(I - \tilde{\Lambda}\right)^{-1} U$ and

$$
\begin{aligned}
Ze\left(-cotg\left(\frac{\theta}{2}\right)\right) &= \begin{pmatrix} D^{-1} + V^*\left(\tilde{F} + cotg\left(\frac{\theta}{2}\right) I\right)^+ V & V^*P \\ P^*V & 0 \end{pmatrix} \\
&= \begin{pmatrix} D^{-1} + R^* \begin{pmatrix} \ddots & & \\ & cotg\left(\frac{\theta}{2}\right) - cotg\left(\frac{\tilde{\theta}_j}{2}\right) & \\ & & \ddots \end{pmatrix}^+ R & R^*E \\ E^*R & 0 \end{pmatrix},
\end{aligned}
$$

where $E = \tilde{Q}^* P$, with P consisting of the columns of Q corresponding to the eigenvalue $e^{i\theta}$. So E^*R consists of the rows of R and thus the computation of the extended matrix $Ze(.)$ is the same as that of $Z(.)$.

If the matrix $Z\left(-cotg\left(\frac{\theta}{2}\right)\right)$ has to be computed for θ close to $\tilde{\theta}_j$, then we have overflow problems. In this case we compute

$$
\det Z(\lambda)\,(\tilde{\lambda}_{j-1} - \lambda)^{\mu_{j-1}}\,(\tilde{\lambda}_j - \lambda)^{\mu_j}, \qquad \tilde{\lambda}_{j-1} < \lambda < \tilde{\lambda}_j,
$$

where $\lambda = -cotg\left(\frac{\theta}{2}\right)$, $\tilde{\lambda}_j = -cotg\left(\frac{\tilde{\theta}_j}{2}\right)$.

Since the eigenvectors of F and $G'_s\tilde{M}_n$ are exactly the same, it suffices to explain how we can compute the eigenvectors of F from those of $\tilde{F}$. Computations similar to these in [3] give us the eigenvectors in terms of M_n and $\tilde{M}_n$.

THEOREM 4.7.

(i) *Let $e^{i\theta} \in \sigma(M_n)$, but $e^{i\theta} \notin \sigma(\tilde{M}_n)$, and let $y \in \mathbb{C}^{p\times 1}$ be an element of the null space of $Z(-cotg\left(\frac{\theta}{2}\right))$. Then*

$$
\tilde{Q} \begin{pmatrix} \ddots & & \\ & cotg\left(\frac{\theta}{2}\right) - cotg\left(\frac{\tilde{\theta}_j}{2}\right) & \\ & & \ddots \end{pmatrix}^{-1} Ry
$$

is an eigenvector of $(I - WW^*)\tilde{M}_n$ corresponding to the eigenvalue $e^{i\theta}$, and

$$
\begin{pmatrix} \tilde{Q}_1 & \\ & \tilde{Q}_2\tilde{\Lambda}_2 \end{pmatrix} \begin{pmatrix} \ddots & & \\ & cotg\left(\frac{\theta}{2}\right) - cotg\left(\frac{\tilde{\theta}_j}{2}\right) & \\ & & \ddots \end{pmatrix}^{-1} Ry
$$

is an eigenvector of M_n corresponding to the eigenvalue $e^{i\theta}$.

(ii) *Let $e^{i\theta} \in \sigma(M_n)$, $e^{i\theta} \in \sigma(\tilde{M}_n)$, and let $\begin{pmatrix} y \\ z \end{pmatrix} \in \mathbb{C}^{(p+\mu)\times 1}$ be an element of the null space of $Ze(-cotg\left(\frac{\theta}{2}\right))$. Then*

$$
\tilde{Q} \left[\begin{pmatrix} \ddots & & \\ & cotg\left(\frac{\theta}{2}\right) - cotg\left(\frac{\tilde{\theta}_j}{2}\right) & \\ & & \ddots \end{pmatrix}^+ Ry + Ez \right]
$$

is an eigenvector of $(I - WW^*)\tilde{M}_n$, *corresponding to the eigenvalue* $e^{i\theta}$, *and*

$$\begin{pmatrix} \tilde{Q}_1 & \\ & \tilde{Q}_2\tilde{\Lambda}_2 \end{pmatrix} \left[\begin{pmatrix} \ddots & & \\ & cotg\left(\frac{\theta}{2}\right) - cotg\left(\frac{\tilde{\theta}_j}{2}\right) & \\ & & \ddots \end{pmatrix}^+ Ry + Ez \right]$$

is an eigenvector of M_n *corresponding to the eigenvalue* $e^{i\theta}$.

These transformed vectors are orthonormalized with respect to the classical inner product, but it is also possible to orthonormalize the vectors of the null space with respect to a new inner product:

(i) If $e^{i\theta} \notin \sigma(\tilde{M}_n)$, then we orthonormalize the vectors $\{y_i\}$ of the null space with respect to the inner product

$$\langle y_i, y_j \rangle = y_i^* H^* H y_j,$$

where

$$H = \begin{pmatrix} \ddots & & \\ & cotg\left(\frac{\theta}{2}\right) - cotg\left(\frac{\tilde{\theta}_j}{2}\right) & \\ & & \ddots \end{pmatrix}^{-1} R.$$

(ii) If $e^{i\theta} \in \sigma(\tilde{M}_n)$, then we orthonormalize the vectors $\{\begin{pmatrix} y_i \\ z_i \end{pmatrix}\}$ of the null space with respect to the inner product

$$\langle \begin{pmatrix} y_i \\ z_i \end{pmatrix}, \begin{pmatrix} y_j \\ z_j \end{pmatrix} \rangle = (y_i^* \ z_i^*) \begin{pmatrix} H^*H & 0 \\ 0 & I_\mu \end{pmatrix} \begin{pmatrix} y_j \\ z_j \end{pmatrix},$$

where μ is the multiplicity of $e^{i\theta}$ as eigenvalue of $\tilde{M}_n$, and where

$$H = \begin{pmatrix} \ddots & & \\ & cotg\left(\frac{\theta}{2}\right) - cotg\left(\frac{\tilde{\theta}_j}{2}\right) & \\ & & \ddots \end{pmatrix}^{+} R.$$

Afterwards these vectors are transformed. This method is more suitable if we only need the first p components of the orthonormalized eigenvectors.

We already mentioned that the method described above cannot be used if $1 \in \sigma(\tilde{M}_n)$ or $1 \in \sigma(M_n)$. Also we get problems if we have to compute $cotg\left(\frac{\theta}{2}\right)$, with $|\theta|$ or $|2\pi - \theta|$ small (In the examples we have taken $< 10^{-3}$). Both problems can be avoided by rotating the spectra over an angle ϕ. We will choose ϕ positive.

Since we know the spectrum of $\tilde{M}_n$, it is not a problem to check whether $e^{i\phi}$ belongs to this spectrum. To check whether $e^{i\phi}$ belongs to the spectrum of M_n, we use the following theorem.

THEOREM 4.8. *Let* $\lambda \notin \sigma(\tilde{M}_n)$. *Then* $\det((I - WW^*)\tilde{M}_n - \lambda I) = 0$ *if and only if* $\det(I_p - W^*\tilde{M}_n(\tilde{M}_n - \lambda I)^{-1}W) = 0$.

Proof. Let

$$
\begin{aligned}
M_r &:= \begin{pmatrix} I_{np} & 0 \\ -W^*\tilde{M}_n & I_p \end{pmatrix} \begin{pmatrix} I_{np} & (\tilde{M}_n-\lambda I)^{-1}W \\ 0 & I_p \end{pmatrix} \\
&= \begin{pmatrix} I_{np} & (\tilde{M}_n-\lambda I)^{-1}W \\ -W^*\tilde{M}_n & I_p - W^*\tilde{M}_n(\tilde{M}_n-\lambda I)^{-1}W \end{pmatrix}
\end{aligned}
$$

and

$$
\begin{aligned}
M_l &:= \begin{pmatrix} I_{np} & 0 \\ W^*\tilde{M}_n(\tilde{M}_n-\lambda I)^{-1} & I_p \end{pmatrix} \begin{pmatrix} I_{np} & -W \\ 0 & I_p \end{pmatrix} \\
&= \begin{pmatrix} I_{np} & -W \\ W^*\tilde{M}_n(\tilde{M}_n-\lambda I)^{-1} & I_p - W^*\tilde{M}_n(\tilde{M}_n-\lambda I)^{-1}W \end{pmatrix}.
\end{aligned}
$$

Then

$$
\begin{aligned}
& M_l \begin{pmatrix} (I-WW^*)\tilde{M}_n-\lambda I & 0 \\ 0 & I_p \end{pmatrix} M_r \\
&= \begin{pmatrix} \tilde{M}_n-\lambda I & 0 \\ 0 & I_p - W^*\tilde{M}_n(\tilde{M}_n-\lambda I)^{-1}W \end{pmatrix}.
\end{aligned}
$$

Since $\det M_l = 1$ and $\det M_r = 1$ we have $\det((I-WW^*)\tilde{M}_n-\lambda I) = 0$ if and only if $\det(I_p - W^*\tilde{M}_n(\tilde{M}_n-\lambda I)^{-1}W) = 0$. □

Thus, we have to check whether the matrix

$$
U^* \begin{pmatrix} \ddots & & \\ & cotg\left(\frac{\tilde{\theta}_j-\phi}{2}\right) & \\ & & \ddots \end{pmatrix} U
$$

is non-singular.

Suppose the arguments of the eigenvalues of $M_n^{(1)}$ and $M_n^{(2)}$ are combined and sorted: $0 \le \tilde{\theta}_1 \le \ldots \le \tilde{\theta}_m < 2\pi$. First we check whether $\tilde{\theta}_1$ and $\tilde{\theta}_m$ are not in the neighbourhood of zero. If this is not the case and 0 does not belong to the spectrum of M_n, no rotation is needed.

Otherwise additional work is required to avoid problems when computing $cotg\left(\frac{\tilde{\theta}_j}{2}\right)$. If the middle point of the interval $[\tilde{\theta}_m, \tilde{\theta}_1]$ corresponds to a point in the upper half plane, we determine the length of this interval; otherwise we determine the length of the interval $[\tilde{\theta}_1, \tilde{\theta}_2]$. If this length is large enough, then we check whether the middle point belongs to the spectrum of M_n. If not, we take ϕ equal to this middle point. In the other case we repeat the procedure for the neighbouring interval (counter clockwise).

When we have found an angle ϕ, we subtract this value from all the arguments of the eigenvalues of $\tilde{M}_n$. If necessary we adapt these values so that all the arguments are in the interval $[0, 2\pi)$.

Note that this method does not work in all cases, e.g. if the eigenvalues of M_n are exactly in the middle of the intervals formed by the arguments of the eigenvalues of $\tilde{M}_n$ and 1 belongs to one of the spectra, the above described procedure would not work. In such a case another method has to be used. Instead of dividing the intervals into two parts and checking if the middle point satisfies the conditions we asked for, we can divide the interval in three parts and check if one of the new points satisfies the conditions. On the other hand, information about the location of the eigenvalues can help us to choose a proper method.

Bringing everything together gives us the following algorithm to determine the eigensystem of M_n out of the spectral decompositions $\tilde{M}_n^{(i)} = \tilde{Q}_i\,\tilde{\Lambda}_i\,\tilde{Q}_i^*$, $i = 1, 2$.

1. Combine the eigenvalues of $\tilde{M}_n^{(1)}$ and $\tilde{M}_n^{(2)}$, determine and sort the arguments.
2. Let m be the number of elements in $\sigma(\tilde{M}_n)$ and let $0 \leq \tilde{\theta}_1 \leq \ldots \leq \tilde{\theta}_m < 2\pi$.
3. Compute $U = \begin{pmatrix} Q_1^* & 0 \\ 0 & Q_2^* \end{pmatrix} W$ (taking into account that W is sparse).
4. Determine $\phi \geq 0$ such that $1 \notin \sigma(e^{-i\phi}\tilde{M}_n)$, $1 \notin \sigma(e^{-i\phi}M_n)$ and no eigenvalues of $e^{-i\phi}\tilde{M}_n$ are in the neighbourhood of zero.
5. Perform the rotations of the eigenvalues of $\tilde{M}_n$.
6. Compute $R = (I - \tilde{\Lambda})^{-1}U$.
7. For $j = 1, \ldots, m$, check whether $e^{i\tilde{\theta}_j}$ belongs to $\sigma(M_n)$:
   (a) Compute $Ze\left(-cotg\left(\frac{\tilde{\theta}_j}{2}\right)\right)$ and determine its inertia (neg_j, nul_j, pos_j).
   (b) If $nul_j > 0$
      – Determine $\mathcal{N}\left(Ze\left(-cotg\left(\frac{\tilde{\theta}_j}{2}\right)\right)\right)$.
      – Determine an orthonormal base of the eigenspace of M_n, corresponding to the eigenvalue $e^{i\tilde{\theta}_j}$.
8. For $j = 0, 1, \ldots, m$, determine the eigenvalues with argument in the interval $(\tilde{\theta}_j, \tilde{\theta}_{j+1})$, where $\tilde{\theta}_0 = 0$ and $\tilde{\theta}_{m+1} = 2\pi$:
   (a) $(neg_0, nul_0, pos_0) = inertia(D^{-1})$
   $(neg_{m+1}, nul_{m+1}, pos_{m+1}) = inertia(D^{-1})$
   (b) Let M be equal to the number of eigenvalues of M_n with argument in the interval $(\tilde{\theta}_j, \tilde{\theta}_{j+1})$.
   (c) If $M > 0$
      – Isolate by means of bisection, using the inertia of the matrix $Z(-cotg\left(\frac{\theta}{2}\right))$, the eigenvalues with argument in the interval.
      – Determine the isolated eigenvalues by using a ‘zerofinder’ on $\det Z(-cotg\left(\frac{\theta}{2}\right)) = 0$.
      – Determine the corresponding eigenvectors.

In the next paragraph we describe an implementation of this algorithm on a distributed memory multiprocessor.

4.2. Implementation of the algorithm. First the data has to be distributed among the different processors. The master processor divides the system into two parts and sends the data to its children. After they have received their part of the data, they compute the singular value decomposition $H_s = UDV^*$ and determine the subsystems and the unitary Hermitian matrix G'_s.

One subsystem $M(H_1, H_2, \ldots, H_{s-1}, R)$ consists of the first $s-1$ arguments of the original system $M(H_1, H_2, \ldots, H_{n-1}, K)$, and the argument $R = UV^*$ can be computed from the singular value decomposition of H_s. For the computation of the other subsystem, namely $M(H_{s+1}R^*, H_{s+2}R^*, \ldots, H_{n-1}R^*, KR^*)$, the products

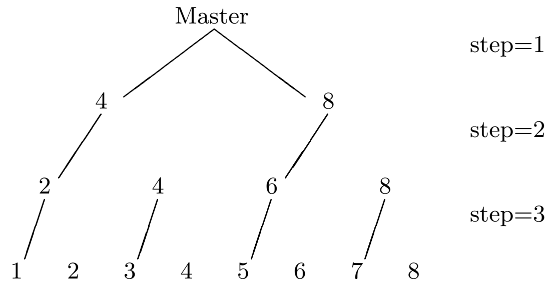
ETNA
Kent State University
etna@mcs.kent.edu

H_iR^* and $R(I - H_i^*H_i)^{1/2}R^*$, $i = s+1, \ldots, n-1$, have to be determined.

The matrix $G_s' = I - W_sW_s^*$ is maintained by storing the 2 non-zero blocks of W_s. One processor computes and stores $W_s^{(1)} = U(I+D)^{1/2}U^*$, while the other one computes and stores $W_s^{(2)} = -U(I-D)^{1/2}U^*$.

The divide procedure is repeated until each processor has its own part of the system. This implies that the number of processors has to be a power of 2, i.e. $nproc = 2^q$, but with the actual configurations this is no problem.

Since communication is very expensive we will try to limit it. From step 2 there is only one subsystem which is explicitly sent to a son, the other subsystem remains in the actual processor. For example, if $nproc = 2^3$, the master processor will send data to proc 4 and proc 8. In the second step proc 4 sends data to proc 2 and proc 8 sends to proc 6. The other subsystems are kept in proc 4 and proc 8 respectively. In step 3 this procedure is repeated. The communication scheme is given by



If we let the processor with number $nproc$ serve additionly as the master processor, then communication in the first step can be halved.

In each step we will send the first subsystem, while the second stays in the actual processor. If the architecture permits overlap between communication and computation, this method implies that the extra amount of work which is needed to determine the second subsystem can be overlapped by the communication for the first part.

From the moment each processor has its own subsystem, the unitary block Hessenberg matrix is computed, and the eigensystem is determined by means of the LAPACK routine. At this point we can also use other methods, e.g., the QR-method for unitary matrices which avoids the computation of the matrix itself and is described in [9] for $p = 1$. Afterwards, each processor determines the arguments of the eigenvalues, sorts them and permutes the eigenvectors. During the remaining part of the program the eigenvalues are characterized by means of their arguments.

In the assembly steps some computations can be performed simultaneously. The computation of the matrix U is performed on the two processors involved in the computation of the eigensystem of the larger matrix and the computations of loop (7), and these of loop (8) are performed in parallel. Each processor involved in the computation of the larger eigensystem treats half of the eigenvalues. Also note that the computation of the matrix R can be performed in parallel, but then some extra communication has to be done. Since this is more expensive, each processor will determine the whole matrix R.

During the next assembly step, we transform the computed argument $(+\phi)$, determine the eigenvalue $\cos\theta + i\sin\theta$ and compute the contribution of this eigenvalue to the partial approximation of the integral. During the other assembly steps, we store the arguments as computed, and afterwards we transform them. All arguments greater than 2π are brougth back in the interval $[0, 2\pi)$. During this adaption step we determine how many arguments are adapted, and during the transmission of the data we make sure the arguments are ordered. This procedure avoids explicitly sorting the arguments, but produces an ordered list which is required by the next assembly step.

As in the real case [13] we will use the LDL* decomposition to compute the inertia of a Hermitian matrix and from the moment we have located one eigenvalue, we use a 'zerofinder', based on bisection, linear and inverse quadratic interpolation, to determine the arguments we are looking for.

4.3. Complexity of the algorithm. In the serial algorithm we distinguish three parts. First we compute the unitary lower block Hessenberg matrix. This can be done in $\approx 2\,p^3\,n^2$ complex arithemetic operations. Next we determine eigenvalues and eigenvectors by means of the LAPACK routine. The number of complex operations is given by $\approx 25\,p^3\,n^3$ (see [8]). Finally the quadrature weights have to be determined. Suppose we have $N = n\,p$ different eigenvalues. Then the computation of a quadrature weight can be done in $2p^2$ complex operations. If F and G are matrix polynomials of degree grf and grg respectively, we get the function evaluations after performing $2p^2(grf+grg)$ complex operations. Of course F and G can be other matrix functions. To get the new partial approximation, 2 matrix products and 1 sum have to be performed. So the total number of complex operations for this third part is given by $n\,p\,\left(4p^3 + 3p^2 + 2p^2(grf + grg)\right)$, and the total number of complex operations for the serial algorithm satisfies $N_{ser} \approx 25\,p^3\,n^3$.

To get an estimate of the number of complex operations for the parallel performance of the divide and conquer method, we suppose we have $nproc = 2^q$ processors, with q the height of the binary tree, and we suppose the length of the different subsystems in each step is equal. Again we can distinguish three parts. First there is the division of the system among the different processors which can be done in $\approx (21p^3 + p^2)\,q + 6p^4\,n\left(\frac{nproc-1}{nproc}\right) + \frac{22}{3}\,p^3 + 3p^2$ complex operations. The LAPACK routine for the computation of the $nproc$ eigensystems is performed and requires $\approx 25\,\frac{p^3\,n^3}{nproc^3} + 2\,\frac{p^3\,n^2}{nproc^2} + g\,\frac{p\,n}{nproc}$ complex operations, where is g is the number of computations to compute a trigonometric function. In the assembly steps we do not take into account the order of the extended matrices $Ze(.)$, and we suppose we need s evaluations of a matrix $Z(.)$ in order to locate an eigenvalue. If we only take into account the term n^3, the number of complex operations for this third part is given by $\approx \frac{1}{14}\left(\frac{nproc^3-8}{nproc^3}\right)p^3\,n^3$. This brings the total number of complex operations for the parallel algorithm to $\approx \left(\frac{25}{nproc^3} + \frac{1}{14}\frac{nproc^3-8}{nproc^3}\right)p^3\,n^3$.

The upper bounds for the speed-up of the parallel divide and conquer method with respect to the serial algorithm using the LAPACK routine satisfy

$$S_{nproc} = \frac{N_{ser}}{N_{par}} \approx \frac{25\,p^3\,n^3\,nproc^3}{\left(25 + \frac{1}{14}(nproc^3 - 8)\right)p^3\,n^3} = \frac{350\,nproc^3}{nproc^3 + 342}.$$

Of course these are upper bounds since we only take into account the terms of n^3. Also we do not bring into account the communication. We can show that the number of messages satisfies $N_{mess} = 4\,nproc + 2$ and the total length of the messages, expressed in bytes, is given by

$$\begin{aligned} N_{len} &= 16\frac{nproc-2}{nproc}p^2\,n^2 + n\left(24(q-1)p^2 + 4c(4p^2+3p) + 56p^2 - 4p\right) \\ &+ 16\,nproc\,p^3 + 48\,p^2. \end{aligned}$$

ETNA
Kent State University
etna@mcs.kent.edu

The ratio $\frac{N_{len}}{N_{par}} = \frac{16\,(nproc-2)\,nproc^2}{25\,p\,n}$ shows that n and p have to be large enough to achieve satisfactory speed-up.

4.4. Example. The results we get are quite similar to these of the real case (see [13]). For example $p = 2$, $W_n = I$ and consider the matrix weight function

$$W(\theta) = \begin{pmatrix} 1 & \cos\theta \\ \cos\theta & 1 \end{pmatrix}.$$

In order to get an idea of the errors, we take $F = G = I$. This gives the following relative errors

n	Serial	$nproc = 2$	$nproc = 4$	$nproc = 8$
50	0.18837E-14	0.34917E-14	0.26711E-11	0.56827E-11
100	0.14999E-14	0.60959E-14	0.32993E-12	0.38065E-11
150	0.12240E-14	0.45497E-14	0.11105E-02	0.11106E-02
200	0.56191E-14	0.48696E-14	0.53183E-11	0.40449E-06
250	0.32259E-14	0.21086E-14	0.61844E-09	0.82521E-02
300	0.41379E-14	0.13186E-13	0.39806E-08	0.11105E-02
350	0.49843E-14	0.50799E-12	0.19537E-05	0.19537E-05
400	0.42870E-14	0.18732E-12	0.12672E-07	0.19110E-07

If we take the same weight function, but

$$W_n = \begin{pmatrix} \frac{\sqrt{3}}{2} & \frac{1}{2} \\ -\frac{1}{2} & \frac{\sqrt{3}}{2} \end{pmatrix},$$

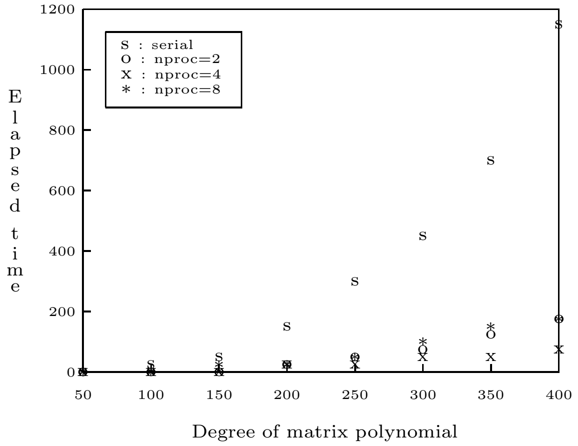
the para-orthogonal matrix polynomials and the corresponding lower block Hessenberg matrix change. So we get different quadrature points and weights. In this case the relative errors are given by

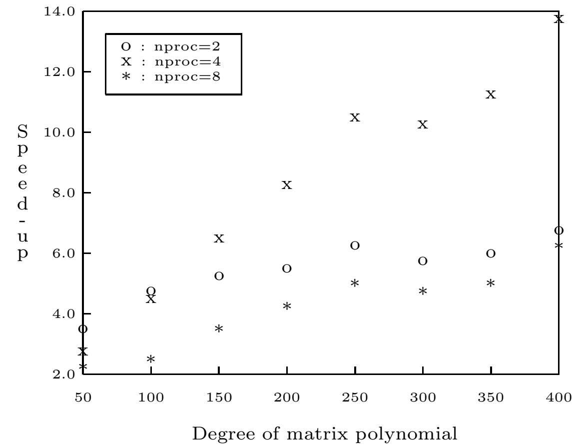
n	Serial	$nproc = 2$	$nproc = 4$
50	0.17135E-14	0.78051E-13	0.33033E-11
100	0.41476E-14	0.56785E-12	0.84741E-12
150	0.19120E-14	0.20545E-12	0.11105E-02
200	0.46207E-14	0.41349E-11	0.92175E-11
250	0.46015E-14	0.55666E-12	0.61858E-09
300	0.27673E-14	0.19644E-10	0.39795E-08
350	0.75841E-14	0.89610E-10	0.19569E-05
400	0.55404E-14	0.18533E-11	0.12673E-07

In both cases we notice the same problem: if the number of processors increases, the number of assembly steps increases and also the relative errors increase. Theoretically there are no problems with multiple eigenvalues or with eigenvalues with very little seperation, but in practice, errors arise when computing the inertia and determinants of the Weinstein matrices Z(.) and Ze(.) (see [2], [3]). If this problem can be fixed, the algorithm can have a practical value.

The execution times and the corresponding speed-ups are given by

ETNA
Kent State University
etna@mcs.kent.edu





Since the upper bound for the speed-up on 2 processors is equal to 8, the timing and corresponding speed-up are execellent. On 4 processors we are much farther away from the maximum, but nevertheless we get a good speed-up. The speed-up on 8 processors is very poor, and the elapsed time is more than that of 2 processors. Performing some analysis about the time spent in communication and computation will give an explanation for the poor results when using 8 processors. It is possible to show that the number of complex operations performed in one second is approximately 11.5, 9.0, 3.0, 0.4 for 1, 2, 4 and 8 processors respectively. This significant decrease of operations performed in one second is due to the fact that we have only counted the number of operations (sum, subtraction, product, division) and did not bring into account the large number of tests which require much more time than an operation. This is also the reason why the results are so poor for a large number of processors.

Acknowledgments. I would like to thank W. Van Assche for his careful reading of this manuscript.

REFERENCES

[1] G.S. Ammar, L. Reichel and D.C. Sorensen, *An implementation of a divide and conquer algorithm for the unitary eigenproblem*, ACM Trans. Math. Softw., 18 (1992), pp. 292-307.

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

[3] P. Arbenz and G.H. Golub, *On the spectral decomposition of hermitian matrices modified by low rank perturbations with applications*, SIAM J. Matrix Anal. Appl., 9 (1988), pp. 40–58.

[4] C.F. Borges and W.B. Gragg, *A parallel divide and conquer algorithm for the generalized real symmetric definite tridiagonal eigenproblem*, in *Numerical Linear Algebra*, L. Reichel, A. Ruttan, R.S. Varga, eds., de Gruyter, Berlin, 1993, pp. 11-29.

[5] Ph. Delsarte, Y. Genin and Y. Kamp, *Orthogonal polynomial matrices on the unit circle*, IEEE Trans. Circuits Systems I Fund. Theory Appl., 25 (1978), pp. 149–160.

[6] J.S. Geronimo, *Matrix orthogonal polynomials on the unit circle*, J. Math. Phys., 22 (1981), pp. 1359–1365.

[7] I. Gohberg, P. Lancaster, L. Rodman, *Matrix Polynomials*, Academic Press, New York, 1982.

[8] G.H. Golub and C.F. Van Loan, *Matrix Computations (2nd edition)*, The John Hopkins University Press, Baltimore, 1991.

[9] W.B. Gragg, *The QR algorithm for unitary Hessenberg matrices*, J. Comput. Appl. Math., 16 (1986), pp. 1-8.

[10] W.B. Gragg and L. Reichel, *A divide and conquer method for the unitary eigenproblem*, in *Hypercube Multiprocessors*, M.T.Heath, ed. , SIAM, Philadelphia, 1987, pp. 639-647.

[11] W.B. Gragg and L. Reichel, *A divide and conquer method for unitary and orthogonal eigenproblems*, Numer. Math., 57 (1990), pp. 695-718.

[12] W.B. Jones, O. Njåstad, W.J. Thron, *Moment theory, orthogonal polynomials, quadrature and continued fractions associated with the unit circle*, Bull. London Math. Soc., 21 (1989), pp. 113-152.

[13] A. Sinap, *Gaussian quadrature for matrix–valued functions on the real line*, to appear in J. Comput. Appl. Math.

[14] A. Sinap and W. Van Assche, *Polynomial interpolation and Gaussian quadrature for matrix-valued functions*, Linear Algebra Appl., 207 (1994), pp. 71–114.

[15] A. Sinap and W. Van Assche, *Orthogonal matrix polynomials and applications*, to appear in J. Comput. Appl. Math.