COMPUTATION OF GAUSS-KRONROD QUADRATURE RULES

D. CALVETTI, G. H. GOLUB, W. B. GRAGG, AND L. REICHEL

ABSTRACT. Recently Laurie presented a new algorithm for the computation of $(2n+1)$-point Gauss-Kronrod quadrature rules with real nodes and positive weights. This algorithm first determines a symmetric tridiagonal matrix of order $2n+1$ from certain mixed moments, and then computes a partial spectral factorization. We describe a new algorithm that does not require the entries of the tridiagonal matrix to be determined, and thereby avoids computations that can be sensitive to perturbations. Our algorithm uses the consolidation phase of a divide-and-conquer algorithm for the symmetric tridiagonal eigenproblem. We also discuss how the algorithm can be applied to compute Kronrod extensions of Gauss-Radau and Gauss-Lobatto quadrature rules. Throughout the paper we emphasize how the structure of the algorithm makes efficient implementation on parallel computers possible. Numerical examples illustrate the performance of the algorithm.

1. INTRODUCTION

Let dw be a nonnegative measure on the real interval $[a, b]$ with an infinite number of points of increase, and such that the moments $\mu_k := \int_a^b x^k dw(x)$, $k = 0, 1, 2, \ldots$, exist and are bounded. For notational convenience, we assume that $\mu_0 = 1$. An n-point Gauss quadrature rule for the integral

$$(1.1) \qquad \mathcal{I}f := \int_a^b f(x)dw(x)$$

is a formula of the form

$$(1.2) \qquad \mathcal{G}_nf := \sum_{k=1}^n f(x_k)w_k$$

with the nodes $a < x_1 < x_2 < \cdots < x_n < b$ and positive weights w_k chosen so that

$$(1.3) \qquad \mathcal{G}_nf = \mathcal{I}f \qquad \forall f \in \mathbb{P}_{2n-1}.$$

Here and throughout this paper $\mathbb{P}_j$ denotes the set of polynomials of degree at most j. The associated Gauss-Kronrod quadrature rule

$$(1.4) \qquad \mathcal{K}_{2n+1} f := \sum_{k=1}^{2n+1} f(\tilde{x}_k) \tilde{w}_k$$

has the properties that

$$(1.5) \qquad \{x_k\}_{k=1}^n \subset \{\tilde{x}_k\}_{k=1}^{2n+1}$$

and

$$(1.6) \qquad \mathcal{K}_{2n+1} f = \mathcal{I} f \qquad \forall f \in \mathbb{P}_{3n+1}.$$

We present a new algorithm for the computation of Gauss-Kronrod quadrature rules with real nodes and positive weights when such rules exist. Our algorithm is based on recent results by Laurie [12] on properties of symmetric tridiagonal matrices associated with Gauss-Kronrod rules.

In typical applications of Gauss-Kronrod quadrature rules, both $\mathcal{G}_n f$ and $\mathcal{K}_{2n+1} f$ are evaluated, and this pair of approximations of $\mathcal{I} f$ is used to estimate the error in $\mathcal{G}_n f$. Applications in adaptive quadrature routines can be computationally demanding, and therefore it is important to develop accurate and fast algorithms that are well suited for implementation on a parallel computer for the computation of nodes and weights of Gauss-Kronrod rules; see [4, 8, 15] for recent discussions. Surveys of properties of Gauss-Kronrod quadrature rules are presented by Gautschi [7], Laurie [12] and Monegato [14]; see also Golub and Kautsky [9] for related discussions.

Let $\{p_j\}_{j=0}^\infty$ be a sequence of monic orthogonal polynomials with respect to the inner product

$$(1.7) \qquad (f, g) := \int_a^b f(x) g(x) dw(x),$$

i.e.,

$$(1.8) \qquad (p_j, p_k) = 0, \qquad j \neq k.$$

The p_j satisfy the recursion relations

$$(1.9) \qquad \begin{aligned} p_{k+1}(x) &= (x - a_k) p_k(x) - b_k^2 p_{k-1}(x), \qquad k = 1, 2, \dots, \\ p_1(x) &:= x - a_0, \qquad p_0(x) := 1, \end{aligned}$$

with coefficients

$$(1.10) \qquad a_k := \frac{(p_k, x p_k)}{(p_k, p_k)}, \qquad k = 0, 1, \dots,$$

$$(1.11) \qquad b_k^2 := \frac{(p_k, p_k)}{(p_{k-1}, p_{k-1})}, \qquad k = 1, 2, \dots .$$

Note that $(p_0, p_0) = \mu_0 = 1$. It follows from (1.11) that

$$(1.12) \qquad (p_k, p_k) = b_k^2 b_{k-1}^2 \cdots b_1^2, \qquad k \geq 1.$$

Define the positive quantities $b_k := (b_k^2)^{1/2}$, $k \geq 1$. We refer to the a_k and b_k as recursion coefficients for the family of orthogonal polynomials (1.9). The $2n - 1$

coefficients $\{a_k\}_{k=0}^{n-1}$ and $\{b_k\}_{k=1}^{n-1}$ determine the symmetric tridiagonal matrix

$$(1.13) \qquad T_n := \begin{bmatrix} a_0 & b_1 & & & & \\ b_1 & a_1 & b_2 & & & \\ & \ddots & \ddots & \ddots & & \\ & & b_{n-2} & a_{n-2} & b_{n-1} & \\ & & & b_{n-1} & a_{n-1} \end{bmatrix} \in \mathbb{R}^{n \times n}$$

with spectral factorization

$$(1.14) \qquad T_n = W_n \Lambda_n W_n^T, \qquad \Lambda_n = \text{diag}[\lambda_1, \lambda_2, \ldots, \lambda_n], \qquad W_n W_n^T = I.$$

Due to the positivity of the off-diagonal entries b_k, the eigenvalues λ_j are distinct and all entries of the first row of W_n are nonvanishing. Moreover, it is well known that the nodes and weights of the Gauss rule (1.2) are given by

$$(1.15) \qquad \begin{cases} x_j := \lambda_j, \\ w_j := (e_1^T W_n e_j)^2, \end{cases} \qquad 1 \le j \le n,$$

where e_j denotes the jth axis vector; see, e.g., [10]. We refer to the sets of eigenvalues and first or last components of normalized eigenvectors of a matrix as partial spectral resolution of the matrix. The sets $\{\lambda_j\}_{j=1}^n \cup \{e_1^T W_n e_j\}_{j=1}^n$ and $\{\lambda_j\}_{j=1}^n \cup \{e_n^T W_n e_j\}_{j=1}^n$ are partial spectral resolutions of T_n. We will assume that the recursion coefficients a_j and b_j are available. The nodes and weights (1.15) of the Gauss rule can then be computed in $\mathcal{O}(n^2)$ arithmetic operations by the Golub-Welsch algorithm [10].

Our algorithm for the computation of the nodes and weights of the Gauss-Kronrod rule (1.4) requires that the last entries of the normalized eigenvectors of T_n also be available. These can be computed simultaneously with the Gauss weights by modifying the Golub-Welsch algorithm in a straightforward manner. The operation count for the modified algorithm is also $\mathcal{O}(n^2)$. The eigenvalues and first and last components of normalized eigenvectors can also conveniently be determined by one of the divide-and-conquer algorithms for the symmetric tridiagonal eigenproblem presented by Borges and Gragg [3] or Gu and Eisenstat [11]. These algorithms also require $\mathcal{O}(n^2)$ arithmetic operations, and with n processors the computations can be carried out in $\mathcal{O}(n)$ time steps.

Laurie [12] pointed out that if the Gauss-Kronrod rule (1.4) has distinct real nodes $\{\tilde{x}_k\}_{k=1}^{2n+1}$ and positive weights $\{\tilde{w}_k\}_{k=1}^{2n+1}$, then there is an associated symmetric tridiagonal matrix

$$(1.16) \qquad \tilde{T}_{2n+1} := \begin{bmatrix} \tilde{a}_0 & \tilde{b}_1 & & & & \\ \tilde{b}_1 & \tilde{a}_1 & \tilde{b}_2 & & & \\ & \ddots & \ddots & \ddots & & \\ & & \tilde{b}_{2n-1} & \tilde{a}_{2n-1} & \tilde{b}_{2n} & \\ & & & \tilde{b}_{2n} & \tilde{a}_{2n} \end{bmatrix} \in \mathbb{R}^{(2n+1) \times (2n+1)},$$

with spectral factorization

$$\tilde{T}_{2n+1} = \tilde{W}_{2n+1} \tilde{\Lambda}_{2n+1} \tilde{W}_{2n+1}^T, \quad \tilde{\Lambda}_{2n+1} = \text{diag}[\tilde{\lambda}_1, \tilde{\lambda}_2, \ldots, \tilde{\lambda}_{2n+1}], \quad \tilde{W}_{2n+1} \tilde{W}_{2n+1}^T = I,$$

such that

$$(1.17) \qquad \begin{cases} \tilde{x}_j = \tilde{\lambda}_j, \\ \tilde{w}_j = (e_1^T \tilde{W}_{2n+1} e_j)^2, \end{cases} \qquad 1 \le j \le 2n+1,$$

analogously to (1.15). We refer to the matrix (1.16) as the Gauss-Kronrod matrix. Let the nodes be ordered according to $\tilde{x}_1 < \tilde{x}_2 < \cdots < \tilde{x}_{2n+1}$. Monegato [13] showed that the positivity of the weights $\tilde{w}_k$ associated with nodes $\tilde{x}_k \notin \{x_j\}_{j=1}^n$ is equivalent with the interlacing property

$$\tilde{x}_1 < \tilde{x}_2 = x_1 < \tilde{x}_3 < \tilde{x}_4 = x_2 < \tilde{x}_5 < \cdots < \tilde{x}_{2n} = x_n < \tilde{x}_{2n+1}.$$

Proposition 1.1 (Laurie [12]). *Let $\hat{T}_n$ and $\check{T}_n$ denote the leading and trailing $n \times n$ principal submatrices of $\tilde{T}_{2n+1}$, respectively. Then $\hat{T}_n$ and $\check{T}_n$ have the same eigenvalues. Moreover, for n odd,*

$$(1.18) \qquad \tilde{a}_{j-1} = a_{j-1}, \qquad \tilde{b}_j = b_j, \qquad 1 \le j \le \frac{3n+1}{2},$$

and, for n even,

$$(1.19) \qquad \begin{cases} \tilde{a}_j = a_j, & 0 \le j \le \frac{3n}{2}, \\ \tilde{b}_j = b_j, & 1 \le j \le \frac{3n}{2}. \end{cases}$$

Proof. Formulas (1.18) and (1.19) express that the first $3n+1$ coefficients of the matrices $\tilde{T}_{2n+1}$ and T_{2n+1} agree. This result follows immediately from (1.6). In particular, $\hat{T}_n = T_n$. This observation and the fact that $\{\lambda_j\}_{j=1}^n \subset \{\tilde{\lambda}_j\}_{j=1}^{2n+1}$ implies that $\hat{T}_n$ and $\check{T}_n$ have the same spectrum, as can be seen by expanding $\det(\tilde{T}_{2n+1} - \lambda I)$ along the $(n+1)$st row; see [12] for details. $\qquad \square$

It follows from Proposition 1.1 that the existence of a Gauss-Kronrod quadrature rule with real distinct nodes and positive weights is equivalent to the existence of a real solution to the following inverse eigenvalue problem.

Corollary 1.2. *Let the first $n-1$ entries of the $n \times n$ symmetric tridiagonal matrix*

$$(1.20) \qquad \check{T}_n := \begin{bmatrix} \tilde{a}_{n+1} & \tilde{b}_{n+2} & & & & \\ \tilde{b}_{n+2} & \tilde{a}_{n+2} & \tilde{b}_{n+3} & & & \\ & \ddots & \ddots & \ddots & & \\ & & \tilde{b}_{2n-1} & \tilde{a}_{2n-1} & \tilde{b}_{2n} \\ & & & \tilde{b}_{2n} & \tilde{a}_{2n} \end{bmatrix}$$

be determined by (1.18) when n is odd, and by (1.19) when n is even. Let the eigenvalues of $\check{T}_n$ be the eigenvalues of the matrix (1.13). There is a real symmetric tridiagonal matrix $\check{T}_n$ with these properties if and only if there is a $(2n+1)$-point Gauss-Kronrod quadrature rule (1.4) with real nodes and positive weights.

Example 1.1. Let $n = 1$. The entries $\{\tilde{a}_j\}_{j=0}^1$ and $\{\tilde{b}_j\}_{j=1}^2$ of the Gauss-Kronrod matrix $\tilde{T}_3$ are recursion coefficients for the orthogonal polynomials associated with the measure dw. The entry marked by $*$ is not explicitly known,

$$\tilde{T}_3 := \begin{bmatrix} \tilde{a}_0 & \tilde{b}_1 & \cdot \\ \tilde{b}_1 & \tilde{a}_1 & \tilde{b}_2 \\ & \tilde{b}_2 & * \end{bmatrix}$$

However, by Proposition 1.1, $\tilde{a}_2 = \tilde{a}_0$. In particular, any 3-point Gauss-Kronrod rule associated with a 1-point Gauss rule has real nodes and positive weights. $\qquad \square$

Example 1.2. Let $n = 2$. The entries $\{\tilde{a}_j\}_{j=0}^3$ and $\{\tilde{b}_j\}_{j=1}^3$ of the Gauss-Kronrod matrix $\tilde{T}_5$ are recursion coefficients for orthogonal polynomials associated with the measure dw, but the entries marked by $*$ are not explicitly known,

$$(1.21) \qquad \tilde{T}_5 := \begin{bmatrix} \tilde{a}_0 & \tilde{b}_1 & & & \\ \tilde{b}_1 & \tilde{a}_1 & \tilde{b}_2 & & \\ & \tilde{b}_2 & \tilde{a}_2 & \tilde{b}_3 & \\ & & \tilde{b}_3 & \tilde{a}_3 & * \\ & & & * & * \end{bmatrix}$$

By Proposition 1.1 the leading and trailing principal 2×2 submatrices of $\tilde{T}_5$ have the same trace. This yields the equation

$$(1.22) \qquad \tilde{a}_0 + \tilde{a}_1 = \tilde{a}_3 + \tilde{a}_4$$

for $\tilde{a}_4$. The determinants of the leading and trailing principal 2×2 submatrices are also the same, and this gives the equation

$$(1.23) \qquad \tilde{a}_0\tilde{a}_1 - \tilde{b}_1^2 = \tilde{a}_3\tilde{a}_4 - \tilde{b}_4^2$$

for $\tilde{b}_4$. When (1.23) is satisfied by a real positive value of $\tilde{b}_4$, a Gauss-Kronrod rule with real nodes and positive weights exists. $\qquad \square$

Example 1.3. Let $[a, b] = [-1, 1]$ and $dw(x) := \frac{2}{\pi}(1 - x^2)^{1/2}dx$. Then the Gauss-Kronrod matrix (1.21) has the known entries

$$\tilde{T}_5 := \begin{bmatrix} 0 & 1/2 & & & \\ 1/2 & 0 & 1/2 & & \\ & 1/2 & 0 & 1/2 & \\ & & 1/2 & 0 & * \\ & & & * & * \end{bmatrix}$$

Equations (1.22) and (1.23) yield $a_4 = 0$ and $b_4 = 1/2$. The eigenvalues and eigenvectors of this matrix are explicitly known, and we obtain the 5-point Gauss-Kronrod rule

$$\tilde{x}_k = \cos(\frac{\pi}{6}k), \qquad \tilde{w}_k = \frac{1}{3}\sin^2(\frac{\pi}{6}k), \qquad 1 \leq k \leq 5.$$

$\qquad \square$

This paper describes a new algorithm for computing Gauss-Kronrod quadrature rules with real nodes and positive weights. The algorithm first determines the eigenvalues as well as the first and last components of normalized eigenvectors of the matrix T_n. This yields, in particular, the Gauss quadrature rule (1.15). The algorithm then proceeds by computing the first components of normalized eigenvectors of the matrix $\tilde{T}_n$ defined in Proposition 1.1. This is described in Section 2. When n is even, we use a method proposed by Boley and Golub [2]. For n odd, we apply a closely related method. We remark that our algorithm does not explicitly determine the tridiagonal matrix (1.20). After these initial calculations, a consolidation step of the divide-and-conquer algorithm presented by Borges and Gragg [3] is used to determine the eigenvalues and first components of normalized eigenvectors of the matrix (1.16), and by (1.17) we obtain the Gauss-Kronrod rule. Relevant details of the divide-and-conquer method are discussed in Section 3. Our algorithm determines the $(2n + 1)$-point Gauss-Kronrod rule (1.17) from the recursion coefficients a_j and b_j in only $\mathcal{O}(n^2)$ arithmetic operations, and with n

processors only $\mathcal{O}(n)$ time steps are required. Section 4 describes how the algorithm of Section 3 can be applied to the computation of Gauss-Kronrod-Radau and Gauss-Kronrod-Lobatto rules. These rules are Kronrod extensions of Gauss-Radau and Gauss-Lobatto rules, respectively, and find application in adaptive composite quadrature rules. Section 5 contains numerical examples.

When only the measure dw but not the recursion coefficients a_j and b_j are available, the latter can be computed by (1.10) and (1.11). It may be attractive to evaluate necessary inner products by a Clenshaw-Curtis quadrature rule; see Gautschi [6] for a discussion.

Laurie [12] presented another algorithm for the computation of $(2n + 1)$-point Gauss-Kronrod rules in $\mathcal{O}(n^2)$ arithmetic operations. This algorithm first determines certain mixed moments from which the symmetric tridiagonal matrix (1.16) is determined. The Gauss-Kronrod nodes and weights are then determined by applying the Golub-Welsch algorithm to the matrix (1.16).

Our algorithm avoids the explicit determination of the matrix (1.16). Experience from related problems indicates that the computation of the entries of $\tilde{T}_{2n+1}$ can be sensitive to round-off errors; see, e.g., [5].

2. COMPUTATION OF EIGENVECTOR COMPONENTS OF $\check{T}_n$

We consider the determination of the first components of normalized eigenvectors of the real symmetric tridiagonal matrix (1.20), which is the trailing principal $n \times n$ submatrix of the Gauss-Kronrod matrix (1.16). The $n - 1$ first entries of $\check{T}_n$ are given by (1.18) or (1.19). The remaining diagonal and subdiagonal matrix entries are not known. The matrix (1.20) is required to have the same eigenvalues $\lambda_1 < \lambda_2 < \cdots < \lambda_n$ as T_n. We assume for the moment that such a real symmetric tridiagonal matrix $\check{T}_n$ with positive subdiagonal elements exists.

We first outline a method due to Boley and Golub [2] that can be applied when n is even. A modification of this method, described below, can be used for n odd.

Recall that the matrix T_n is associated with a positive measure dw with support in a real interval $[a, b]$ and with the quadrature rule (1.15). Similarly, we may associate with the matrix $\check{T}_n$ a nonnegative measure $d\check{w}$ with support in a real interval $[\check{a}, \check{b}]$ and such that $\int_{\check{a}}^{\check{b}} d\check{w}(x) = 1$. The eigenvalues λ_j and squares of the first components of normalized eigenvectors $\check{w}_j$ define a quadrature rule $\{\lambda_j, \check{w}_j\}_{j=1}^{n}$ associated with the matrix $\check{T}_n$, such that

$$(2.1) \qquad \int_{\check{a}}^{\check{b}} x^k d\check{w} = \sum_{j=1}^{n} \lambda_j^k \check{w}_j, \qquad 0 \le k < n.$$

We remark that we may choose $d\check{w}$ to be the discrete measure defined by the quadrature rule $\{\lambda_j, \check{w}_j\}_{j=1}^{n}$.

Let n be even. Then the entries of the leading principal submatrix $\check{T}_{n/2} \in \mathbb{R}^{n/2 \times n/2}$ are explicitly known. Let $\{x_j^*, w_j^*\}_{j=1}^{n/2}$ be the Gauss quadrature rule associated with the matrix $\check{T}_{n/2}$, i.e., the x_j^* are eigenvalues and the w_j^* are the square of the first components of normalized eigenvectors of $\check{T}_{n/2}$; cf. (1.15). Both quadrature rules $\{x_j^*, w_j^*\}_{j=1}^{n/2}$ and $\{\lambda_j, \check{w}_j\}_{j=1}^{n}$ can be regarded as discretizations of the

measure $d\breve{w}$. Thus,

$$(2.2) \qquad \int_{\breve{a}}^{\breve{b}} x^k d\breve{w}(x) = \sum_{j=1}^{n/2} (x_j^*)^k w_j^* = \sum_{j=1}^{n} \lambda_j^k \breve{w}_j, \qquad 0 \le k < n.$$

The equations (2.2) can be expressed in terms of the Lagrange polynomials

$$(2.3) \qquad \ell_k(x) := \prod_{\substack{j=1 \\ j \ne k}}^{n} \frac{x - \lambda_j}{\lambda_k - \lambda_j}, \qquad 1 \le k \le n,$$

and we obtain

$$(2.4) \qquad \sum_{j=1}^{n/2} \ell_k(x_j^*) w_j^* = \sum_{j=1}^{n} \ell_k(\lambda_j) \breve{w}_j = \breve{w}_k, \qquad 1 \le k \le n.$$

We remark that the equations (2.2) can be formulated as a linear system of equations with a Vandermonde matrix for the weights $\breve{w}_j$. Numerical experiments reported in [5] indicate that the weights $\breve{w}_j$ are computed more accurately by formula (2.4).

We assumed above that a real symmetric tridiagonal matrix $\breve{T}_n$ with positive subdiagonal elements, with given spectrum $\{\lambda_j\}_{j=1}^{n}$ and with a given leading $n/2 \times n/2$ principal submatrix exists. However, this is not always the case. For instance, when $dw(x) = e^{-x} dx$ and $[a, b] = [0, \infty]$, the matrix $\breve{T}_n$ is for many values of n complex symmetric, with real diagonal entries and at least one purely imaginary subdiagonal element. The measure $d\breve{w}$ associated with such a matrix $\breve{T}_n$ is indefinite, and at least one weight $\breve{w}_j$ is negative. A numerical method for computing complex symmetric tridiagonal matrices of even order n with real diagonal entries and real or purely imaginary subdiagonal entries, given its real distinct eigenvalues and its real symmetric tridiagonal leading principal submatrix of order $n/2$, is described in [5].

The present paper is concerned with the computation of Gauss-Kronrod rules with distinct real nodes and positive weights, and by Corollary 1.2 we may restrict our attention to real symmetric tridiagonal matrices $\breve{T}_n$ with positive subdiagonal entries. In particular, we are only concerned with the case when the weights $\breve{w}_j$ are positive.

When n is odd, the algorithm described above has to be modified. The entries $\{\tilde{a}_j\}_{j=n+1}^{(3n-1)/2}$ and $\{\tilde{b}_j\}_{j=n+2}^{(3n+1)/2}$ of $\breve{T}_n$ are known. The largest leading $k \times k$ principal submatrix of $\breve{T}_n$ with all entries explicitly known is of order $k = (n-1)/2$, and the Gauss rule associated with this submatrix is not of high enough order to allow the matching of n moments, analogously to (2.2). Therefore a formula similar to (2.4) cannot be applied before some preliminary calculations have been carried out.

The computations are divided into two steps. First we compute the diagonal entry $\tilde{a}_{(3n+1)/2}$. Then the leading principal submatrix of order $(n+1)/2$ of $\breve{T}_n$ is known, and we can compute the weights $\breve{w}_j$ by a formula analogous to (2.4).

Let $\{\breve{p}_j\}_{j=0}^{(n-1)/2}$ be the first $(n+1)/2$ monic orthogonal polynomials associated with the inner product

$$\langle f, g \rangle := \int_{\breve{a}}^{\breve{b}} f(x) g(x) d\breve{w}(x).$$

These polynomials can be computed from the available recursion coefficients. The desired diagonal entry of $\check{T}_n$ is given by

$$\tilde{a}_{(3n+1)/2} = \frac{\langle \check{p}_{(n-1)/2}, x\check{p}_{(n-1)/2}\rangle}{\langle \check{p}_{(n-1)/2}, \check{p}_{(n-1)/2}\rangle},$$

where

$$\langle \check{p}_{(n-1)/2}, \check{p}_{(n-1)/2}\rangle = \tilde{b}^2_{(3n+1)/2}\tilde{b}^2_{(3n-1)/2}\cdots\tilde{b}^2_{n+2}.$$

Note that

$$(2.5) \qquad \check{p}_n(x) = \prod_{j=1}^{n}(x - \lambda_j)$$

and that $x\check{p}^2_{(n-1)/2}(x) - \check{p}_n(x) \in \mathbb{P}_{n-1}$. The latter polynomial can be written as

$$(2.6)$$

$$x\check{p}^2_{(n-1)/2}(x) - \check{p}_n(x) = \check{p}_{(n-1)/2}(x)\sum_{j=0}^{(n-1)/2}c_{(n-1)/2+j}\check{p}_j(x) + \sum_{j=0}^{(n-3)/2}c_j\check{p}_j(x)$$

for certain coefficients c_k. Integrating (2.6) and using the orthogonality of the polynomials $\check{p}_j$ yields

$$\langle \check{p}_{(n-1)/2}, x\check{p}_{(n-1)/2}\rangle = \langle \check{p}_{(n-1)/2}, \sum_{j=0}^{(n-1)/2}c_{(n-1)/2+j}\check{p}_j\rangle + \langle \sum_{j=0}^{(n-3)/2}c_j\check{p}_j, \check{p}_0\rangle$$

$$= c_{n-1}\langle \check{p}_{(n-1)/2}, \check{p}_{(n-1)/2}\rangle + c_0\langle \check{p}_0, \check{p}_0\rangle,$$

and therefore

$$(2.7) \qquad \tilde{a}_{(3n+1)/2} = c_{n-1} + \frac{c_0}{\tilde{b}^2_{(3n+1)/2}\tilde{b}^2_{(3n-1)/2}\cdots\tilde{b}^2_{n+2}}.$$

It remains to determine the coefficients c_{n-1} and c_0. Note that c_{n-1} is the leading coefficient of the polynomial $x\check{p}^2_{(n-1)/2}(x) - \check{p}_n(x)$ in power form. Straightforward expansion in terms of powers of x yields

$$x\check{p}^2_{(n-1)/2}(x) = x^n - \left(2\sum_{j=n+1}^{(3n-1)/2}\tilde{a}_j\right)x^{n-1} + \mathcal{O}(x^{n-2})$$

and

$$\check{p}_n(x) = x^n - \left(\sum_{j=1}^{n}\lambda_j\right)x^{n-1} + \mathcal{O}(x^{n-2}).$$

Therefore

$$x\check{p}^2_{(n-1)/2}(x) - \check{p}_n(x) = \left(\sum_{j=1}^{n}\lambda_j - 2\sum_{j=n+1}^{(3n-1)/2}\tilde{a}_j\right)x^{n-1} + \mathcal{O}(x^{n-2}).$$

Comparison with (2.6) shows that

$$(2.8) \qquad c_{n-1} = \sum_{j=1}^{n}\lambda_j - 2\sum_{j=n+1}^{(3n-1)/2}\tilde{a}_j.$$

We turn to the computation of the coefficient c_0. Determine the Gauss quadrature rule $\{x_j^*, w_j^*\}_{j=1}^{(n-1)/2}$ associated with the leading principal $\frac{n-1}{2} \times \frac{n-1}{2}$ submatrix

of T_n, all of whose entries are known. We then apply this quadrature rule to the right-hand side and left-hand side of (2.6) to obtain

$$(2.9) \qquad c_0 = - \sum_{j=1}^{(n-1)/2} \check{p}_n(x_j^*) w_j^*,$$

where $\check{p}_n$ is given by (2.5). In the derivation of (2.9), we have used that the nodes x_j^* are the zeros of $\check{p}_{(n-1)/2}$, and that by orthogonality

$$\sum_{j=1}^{(n-1)/2} \check{p}_k(x_j^*) w_j^* = 0, \qquad 1 \le k \le \frac{n-3}{2}.$$

Thus, we can evaluate the coefficient $\tilde{a}_{(3n+1)/2}$ by using formulas (2.7)–(2.9).

The leading principal $\frac{n+1}{2} \times \frac{n+1}{2}$ submatrix of $\check{T}_n$ is now explicitly known, and we can determine the weights $\check{w}_j$ analogously as when n is even. Thus, compute the Gauss rule $\{x_j', w_j'\}_{j=1}^{(n+1)/2}$ associated with the leading principal submatrix of $\check{T}_n$ of order $(n+1)/2$. Analogously to (2.2), we obtain

$$\int_{\check{a}}^{\check{b}} x^k d\check{w}(x) = \sum_{j=1}^{(n+1)/2} (x_j')^k w_j' = \sum_{j=1}^{n} \lambda_j^k \check{w}_j, \qquad 0 \le k < n,$$

which, similarly to (2.4), yields the formula for the weights

$$\check{w}_k = \sum_{j=1}^{(n+1)/2} \ell_k(x_j') w_j', \qquad 1 \le k \le n,$$

where the Lagrange polynomials ℓ_k are given by (2.3).

The computations described in this section require $\mathcal{O}(n^2)$ arithmetic operations and can be carried out in $\mathcal{O}(n)$ time steps by n processors when a divide-and-conquer method is used for computing the required quadrature rules.

3. COMPUTATION OF GAUSS-KRONROD RULES

We assume in this section that the eigenvalues and the last components of normalized eigenvectors of the matrix T_n, as well as the first components $\{\check{w}_j^{1/2}\}_{j=1}^n$ of normalized eigenvectors of the matrix $\check{T}_n$, are available. The computation of these quantities is discussed in the previous sections. Recall that the matrices T_n and $\check{T}_n$ have the same eigenvalues. We are now in a position to apply the consolidation phase of the divide-and-conquer algorithm described in [3] to determine the eigenvalues and first components of normalized eigenvectors of $\tilde{T}_{2n+1}$. The associated Gauss-Kronrod rule is then obtained from (1.17).

The Gauss-Kronrod matrix (1.16) can be written as

$$(3.1) \qquad \tilde{T}_{2n+1} := \begin{bmatrix} T_n & e_n b_n & \\ b_n e_n^T & a_n & b_{n+1} e_1^T \\ & e_1 b_{n+1} & \check{T}_n \end{bmatrix}.$$

The matrix T_n has the spectral factorization

$$\check{T}_n = \check{W}_n \Lambda_n \check{W}_n^T, \qquad \check{W}_n \check{W}_n^T = I,$$

where Λ_n is defined by (1.14). Introduce

$$\tilde{U} := \begin{bmatrix} W_n & & \\ & 1 & \\ & & \breve{W}_n \end{bmatrix} \in \mathbb{R}^{(2n+1) \times (2n+1)}.$$

Then

$$(3.2) \qquad \tilde{U}^T (\tilde{T}_{2n+1} - \lambda I) \tilde{U} = \begin{bmatrix} \Lambda_n - \lambda I & W_n^T e_n b_n & \\ b_n e_n^T W_n & a_n - \lambda & b_{n+1} e_1^T \breve{W}_n \\ & \breve{W}_n^T e_1 b_{n+1} & \Lambda_n - \lambda I \end{bmatrix}.$$

Note that the entries of the vectors $e_n^T W_n$ and $e_1^T \breve{W}_n$ are known. The matrix on the right-hand side is the sum of a diagonal matrix and a Swiss cross

$$\tilde{U}^T (\tilde{T}_{2n+1} - \lambda I) \tilde{U} = \begin{bmatrix} x & & & & \\ & x & & & \\ & & x & & \\ & & & x & \\ & & & & x \end{bmatrix} + \begin{bmatrix} & & x & & \\ & & x & & \\ x & x & x & x & x \\ & & x & & \\ & & x & & \end{bmatrix},$$

which we permute to an arrow matrix by a similarity transformation with the permutation matrix $P^{(n+1)} = [e_1, e_2, \ldots, e_n, e_{n+2}, \ldots, e_{2n}, e_{n+1}] \in \mathbb{R}^{(2n+1) \times (2n+1)}$. Thus,

$$(P^{(n+1)})^T \tilde{U}^T (\tilde{T}_{2n+1} \quad - \quad \lambda I) \tilde{U} P^{(n+1)}$$

(3.3)

$$= \begin{bmatrix} \Lambda_n & & W_n^T e_n b_n \\ & \Lambda_n & \breve{W}_n^T e_1 b_{n+1} \\ b_n e_n^T W_n & b_{n+1} e_1^T \breve{W}_n & a_n \end{bmatrix} - \lambda I.$$

We apply rotation similarity transformations to rows j and $j+n$, for $j = 1, 2, \ldots, n$, in order to annihilate the first n entries of row and column $2n + 1$. This process is sometimes referred to as combo-deflation and yields

(3.4)

$$G^T (P^{(n+1)})^T \tilde{U}^T (\tilde{T}_{2n+1} - \lambda I) \tilde{U} P^{(n+1)} G = \begin{bmatrix} \Lambda_n & & \\ & \Lambda_n & c \\ & c^T & a_n \end{bmatrix} - \lambda I,$$

where the matrix $G \in \mathbb{R}^{(2n+1) \times (2n+1)}$ is made up of the product of the n rotations applied to the matrix (3.3) from the right, and the vector $c = [\gamma_1, \gamma_2, \ldots, \gamma_n] \in \mathbb{R}^n$ consists of the entries in positions $n + 1$ through $2n$ of the vector

$$(G^T [b_n e_n^T W_n, b_{n+1} e_1^T \breve{W}_n, a_n])^T.$$

The right-hand side of (3.4) shows that the matrix $\tilde{T}_{2n+1}$ has the diagonal entries of Λ_n as eigenvalues, and these are the nodes of the Gauss rule (1.2). Thus, the computed nodes of the Gauss and Gauss-Kronrod quadrature rules satisfy (1.5).

The remaining $n+1$ eigenvalues of $\tilde{T}_{2n+1}$ are eigenvalues of the trailing principal $(n + 1) \times (n + 1)$ submatrix of the matrix (3.4), which for $\lambda \notin \{x_j\}_{j=1}^n$ can be factored according to

$$\begin{bmatrix} \Lambda_n & c \\ c^T & a_n \end{bmatrix} - I\lambda = \begin{bmatrix} I & 0 \\ c^T (\Lambda_n - \lambda I)^{-1} & 1 \end{bmatrix} \begin{bmatrix} \Lambda_n - \lambda I & c \\ 0^T & -f(\lambda) \end{bmatrix},$$

where f is the spectral function

$$(3.5) \qquad f(\lambda) := \lambda - a_n + \sum_{j=1}^{n} \frac{\gamma_j^2}{x_j - \lambda}.$$

The $n + 1$ distinct zeros of f are the Gauss-Kronrod nodes $\{\tilde{x}_{2j-1}\}_{j=1}^{n+1}$, and they interlace the Gauss nodes $\{x_j\}_{j=1}^{n}$. A zero finder that yields sequences of cubically and monotonically convergent approximations of the zeros of f is described in [3]. The computation of the eigenvalues requires $\mathcal{O}(n^2)$ arithmetic operations. Given the eigenvalues of T_{2n+1}, the first components of normalized eigenvectors also can be determined in $\mathcal{O}(n^2)$ arithmetic operations; see [3] for details. These components are computed by an approach suggested by Gu and Eisenstat; see [11] and references therein. Only $\mathcal{O}(n)$ time steps are required when n processors are available.

The computations required to compute an n-point Gauss quadrature rule and the associated $(2n + 1)$-point Gauss-Kronrod rule are summarized in the following algorithm.

Algorithm 1. Computation of Gauss and Gauss-Kronrod rules.
- **Input**: n, first $3n + 1$ recursion coefficients $a_0, b_1, a_1, b_2, \ldots$ for orthogonal polynomials associated with a positive measure dw scaled so that $\mu_0 = 1$.
- **Output**: n-point Gauss rule $\{x_j, w_j\}_{j=1}^{n}$ and associated $(2n+1)$-point Gauss-Kronrod rule $\{\tilde{x}_j, \tilde{w}_j\}_{j=1}^{2n}$.
- Compute eigenvalues as well as first and last components of normalized eigenvectors of the tridiagonal matrix T_n. The eigenvalues and first components of the eigenvectors yield the Gauss quadrature rule $\{x_j, w_j\}_{j=1}^{n}$ associated with the measure dw.
- Compute weights $\{\breve{w}_j\}_{j=1}^{n}$ of trailing $n \times n$ principal submatrix $\breve{T}_n$ as described in Section 2.
- The entries of row and column $n + 1$ of the matrix $\tilde{T}_{2n}$, the eigenvalues and last component of normalized eigenvectors of T_n and the square-root of the weights $\{\breve{w}_j\}_{j=1}^{n}$ are used to compute the Gauss-Kronrod rule $\{\tilde{x}_j, \tilde{w}_j\}_{j=1}^{2n}$ by application of a consolidation step of a divide-and-conquer algorithm. $\square$

4. GENERALIZED GAUSS-KRONROD QUADRATURE RULES

This section discusses the computation of Gauss-Kronrod rules with one or two preassigned nodes. We refer to these quadrature rules as Gauss-Kronrod-Radau and Gauss-Kronrod-Lobatto rules, respectively. Properties of these rules are discussed by Gautschi [7].

4.1. Gauss-Kronrod-Radau rules. Let dw be the nonnegative measure introduced in Section 1. An $(n + 1)$-point Gauss-Radau quadrature rule for the integral (1.1) with a fixed node at $x = a$ is a formula of the form

$$(4.1) \qquad \mathcal{G}_{n+1,a} f := \sum_{k=0}^{n} f(x_{k,a}) w_{k,a}$$

with nodes $a = x_{0,a} < x_{1,a} < \cdots < x_{n,a} < b$ and positive weights $w_{k,a}$ chosen so that

$$\mathcal{G}_{n+1,a} f = \mathcal{I} f \qquad \forall f \in \mathbb{P}_{2n}.$$

The associated Gauss-Kronrod-Radau quadrature rule

$$(4.2) \qquad \mathcal{K}_{2n+2,a}f := \sum_{k=0}^{2n+1} f(\tilde{x}_{k,a})\tilde{w}_{k,a}$$

has the properties that

$$\{x_{k,a}\}_{k=0}^{n} \subset \{\tilde{x}_{k,a}\}_{k=0}^{2n+1}$$

and

$$(4.3) \qquad \mathcal{K}_{2n+2,a}f = \mathcal{I}f \qquad \forall f \in \mathbb{P}_{3n+2}.$$

In addition, we would like the weights $\tilde{w}_{k,a}$ to be positive and the nodes $\tilde{x}_{k,a}$ to satisfy $a = \tilde{x}_{0,a} < \tilde{x}_{1,a} < \cdots < \tilde{x}_{2n+1,a}$. The "free" nodes $\{x_{k,a}\}_{k=1}^{n}$ of the Gauss-Radau rule (4.1) are zeros of the nth degree orthogonal polynomial associated with the measure

$$(4.4) \qquad dw'(x) := (x-a)dw(x), \qquad a \le x \le b;$$

see, e.g., [9]. Analogously, Gautschi [7] showed that the nodes $\{\tilde{x}_{k,a}\}_{k=1}^{2n+1}$ of the Gauss-Kronrod-Radau rule (4.2) are nodes of a $(2n+1)$-point Gauss-Kronrod quadrature rule

$$(4.5) \qquad \mathcal{K}'_{2n+1}f = \sum_{k=1}^{2n+1} f(\tilde{x}'_k)\tilde{w}'_k$$

associated with the measure (4.4). We apply Algorithm 1 to compute the nodes $\tilde{x}'_k$ and weights $\tilde{w}'_k$ of (4.5) and thereby the nodes $\tilde{x}_{k,a}$ of the Gauss-Kronrod-Radau rule (4.2). The following proposition shows how to compute the weights $\tilde{w}_{k,a}$ of (4.2) from the weights $\tilde{w}'_k$.

Proposition 4.1. *Let* $\{\tilde{w}'_k\}_{k=1}^{2n+1}$ *be the weights of the Gauss-Kronrod quadrature rule* (4.5) *associated with the measure* (4.4). *The weights* $\tilde{w}_{k,a}$ *of the* $(2n+2)$-*point Gauss-Kronrod-Radau rule* (4.2) *associated with the measure dw are given by*

$$(4.6) \qquad \tilde{w}_{k,a} = \frac{\tilde{w}'_k}{\tilde{x}_{k,a} - a}, \qquad 1 \le k \le 2n+1,$$

$$\tilde{w}_{0,a} = \mu_0 - \sum_{k=1}^{2n+1} \tilde{w}_{k,a}.$$

Proof. Introduce the Lagrange polynomials

$$\ell_k(x) := \prod_{\substack{j=1 \\ j \ne k}}^{2n+1} \frac{x - \tilde{x}_{j,a}}{\tilde{x}_{k,a} - \tilde{x}_{j,a}}$$

and

$$\ell_{k,a}(x) := \ell_k(x)\frac{x-a}{\tilde{x}_{k,a} - a},$$

for $1 \le k \le 2n+1$. It follows from (4.3)–(4.5) that

$$\begin{aligned}
\tilde{w}_{k,a} &= \mathcal{K}_{2n+2,a}\ell_{k,a} = \int_a^b \ell_{k,a}(x)dw(x) \\
&= \frac{1}{\tilde{x}_{k,a} - a}\int_a^b \ell_k(x)dw'(x) = \frac{1}{\tilde{x}_{k,a} - a}\mathcal{K}'_{2n+1}\ell_k = \frac{\tilde{w}'_k}{\tilde{x}_{k,a} - a},
\end{aligned}$$

for $1 \le k \le 2n+1$. The formula for $\tilde{w}_{0,a}$ follows from $\sum_{k=0}^{2n+1}\tilde{w}_{k,a} = \int_a^b dw(x)$. $\qquad\square$

Algorithm 1 requires the first $3n + 1$ recursion coefficients $a'_0, b'_1, a'_1, \ldots$ for the orthogonal polynomials with respect to the measure (4.4). When dw is a measure of Jacobi-type

$$(4.7) \qquad dw(x) := c_0(b - x)^\alpha (x - a)^\beta dx, \qquad a < x < b, \qquad \alpha, \beta > -1,$$

then so is dw', and explicit formulas for the recursion coefficients a'_j and b'_j are available; see, e.g., [16]. The scaling factor c_0, where

$$(4.8) \qquad c_0^{-1} := (b - a)^{\alpha + \beta + 1} B(\alpha + 1, \beta + 1)$$

and B denotes the beta function, secures that $\mu_0 = 1$.

When dw is not of Jacobi-type and recursion coefficients a_j and b_j for orthogonal polynomials associated with the measure dw are available, a scheme by Golub and Kautsky [9] can be used to compute recursion coefficients a'_j and b'_j for orthogonal polynomials associated with the measure dw'. Let the symmetric tridiagonal matrix $T_m \in \mathbb{R}^{m \times m}$ be defined by the first $2m - 1$ recursion coefficients a_j and b_j given by (1.10)–(1.11); cf. (1.13). Compute the Choleski factorization

$$(4.9) \qquad T_m - aI = L_m L_m^T.$$

Then the matrix

$$(4.10)$$

$$T'_m := \begin{bmatrix} a'_0 & b'_1 & & & \\ b'_1 & a'_1 & b_2 & & \\ & \ddots & \ddots & \ddots & \\ & & b'_{m-2} & a'_{m-2} & b'_{m-1} \\ & & & b'_{m-1} & a'_{m-1} \end{bmatrix} := L_m^T L_m + aI + \gamma_m e_m e_m^T,$$

where $\gamma_m := b_m^2 / (e_m^T L_m e_m)^2$, contains the first $2m - 1$ recursion coefficients for orthogonal polynomials associated with the measure dw'; see [9, Theorem 3]. The coefficients a'_j and b'_j are used as input for Algorithm 1.

4.2. Gauss-Kronrod-Lobatto rules. Let dw be the nonnegative measure introduced in Section 1. An $(n+2)$-point Gauss-Lobatto quadrature rule for the integral (1.1) with fixed nodes at $x = a$ and $x = b$ is a formula of the form

$$(4.11) \qquad \mathcal{G}_{n+2,a,b} f := \sum_{k=0}^{n+1} f(x_{k,a,b}) w_{k,a,b}$$

with nodes $a = x_{0,a,b} < x_{1,a,b} < \cdots < x_{n,a,b} < x_{n+1,a,b} = b$ and positive weights $w_{k,a,b}$ chosen so that

$$\mathcal{G}_{n+2,a,b} f = \mathcal{I}f \qquad \forall f \in \mathbb{P}_{2n+1}.$$

The associated Gauss-Kronrod-Lobatto quadrature rule

$$(4.12) \qquad \mathcal{K}_{2n+3,a,b} f := \sum_{k=0}^{2n+2} f(\tilde{x}_{k,a,b}) \tilde{w}_{k,a,b}$$

has the properties that

$$\{x_{k,a,b}\}_{k=0}^{n+1} \subset \{\tilde{x}_{k,a,b}\}_{k=0}^{2n+1}$$

and

$$(4.13) \qquad \mathcal{K}_{2n+3,a,b} f = \mathcal{I}f \qquad \forall f \in \mathbb{P}_{3n+3}.$$

We would like the weights $\tilde{w}_{k,a,b}$ to be positive and the nodes $\tilde{x}_{k,a,b}$ to satisfy

$$a = \tilde{x}_{0,a,b} < \tilde{x}_{1,a,b} < \cdots < \tilde{x}_{2n+1,a,b} < \tilde{x}_{2n+2,a,b} = b.$$

The "free" nodes $\{x_{k,a,b}\}_{k=1}^{n}$ of the Gauss-Lobatto rule (4.11) are zeros of the nth degree orthogonal polynomial associated with the measure

$$(4.14) \qquad dw''(x) := (b - x)(x - a)dw(x), \qquad a \le x \le b;$$

see [9]. Analogously, Gautschi [7] showed that the nodes $\{\tilde{x}_{k,a,b}\}_{k=1}^{2n+1}$ of the Gauss-Kronrod-Lobatto rule (4.12) are nodes of a $(2n + 1)$-point Gauss-Kronrod quadrature rule

$$(4.15) \qquad \mathcal{K}_{2n+1}'' f = \sum_{k=1}^{2n+1} f(\tilde{x}_{k,a,b}) \tilde{w}_k''$$

associated with the measure (4.14). We apply Algorithm 1 to compute the nodes $\tilde{x}_{k,a,b}$ and weights $\tilde{w}_k''$ of (4.15) and thereby the nodes of the Gauss-Kronrod-Lobatto rule (4.12). The following proposition shows how to compute the weights $\tilde{w}_{k,a,b}$ of (4.12) from the weights $\tilde{w}_k''$.

Proposition 4.2. *Let $\{\tilde{w}_k''\}_{k=1}^{2n+1}$ be the weights of the Gauss-Kronrod quadrature rule (4.15) associated with the measure (4.14). The weights $\tilde{w}_{k,a,b}$ of the $(2n + 3)$-point Gauss-Kronrod-Lobatto rule (4.12) associated with the measure dw are given by*

$$(4.16) \qquad \tilde{w}_{k,a,b} = \frac{\tilde{w}_k''}{(\tilde{x}_{k,a,b} - a)(b - \tilde{x}_{k,a,b})}, \qquad 1 \le k \le 2n + 1,$$

$$(4.17) \qquad \tilde{w}_{0,a,b} = \frac{1}{b - a} \left(b\mu_0 - \mu_1 + \sum_{k=1}^{2n+1} \tilde{w}_{k,a,b}(\tilde{x}_{k,a,b} - b) \right),$$

$$(4.18) \qquad \tilde{w}_{2n+2,a,b} = \frac{1}{b - a} \left(\mu_1 - a\mu_0 - \sum_{k=1}^{2n+1} \tilde{w}_{k,a,b}(\tilde{x}_{k,a,b} - a) \right).$$

Proof. Formula (4.16) can be shown similarly to (4.6). Integration of $x - b$ by the rule (4.12) yields

$$\mu_1 - b\mu_0 = \int_a^b (x - b)dw(x) = \sum_{k=0}^{2n+1} \tilde{w}_{k,a,b}(\tilde{x}_{k,a,b} - b),$$

from which (4.17) follows. Similarly, (4.18) is obtained by integrating $x - a$. □

Algorithm 1 requires the $3n + 1$ first recursion coefficients $a_0'', b_1'', a_1'', \ldots$ for the orthogonal polynomials with respect to the measure (4.14). When dw is a measure of Jacobi-type (4.7), explicit formulas for these recursion coefficients are available. Otherwise, we can can determine the coefficients a_j'' and b_j'' from the matrix (4.10) as follows. Compute the Choleski factorization

$$bI - T_m' = L_m'(L_m')^T.$$

Then the matrix

$$
T''_m := \begin{bmatrix}
a''_0 & b''_1 & & & & \\
b''_1 & a''_1 & b''_2 & & & \\
& \ddots & \ddots & \ddots & & \\
& & b''_{m-2} & a''_{m-2} & b''_{m-1} \\
& & & b''_{m-1} & a''_{m-1}
\end{bmatrix}
$$
$$
:= bI - (L'_m)^T L'_m + aI + \gamma'_m e_m e_m^T,
$$

where $\gamma'_m := (b'_m)^2/(e_m^T L_m e_m)^2$, contains the first $2m - 1$ recursion coefficients for orthogonal polynomials associated with the measure dw''; see Golub and Kautsky [9] for details.

5. NUMERICAL EXAMPLES

The computations were carried out on an HP 9000 workstation in double precision arithmetic, i.e., with almost 16 significant digits, and in quadruple precision arithmetic. A Matlab implementation of our divide-and-conquer based algorithm is referred to as "d+c based alg." in the tables. This implementation uses double precision arithmetic and is compared to a Fortran implementation using double precision arithmetic of the algorithm presented by Laurie [12]. Laurie's algorithm is referred to as "mixed moment alg." in the tables. We used a QR algorithm from LAPACK [1] to compute the Gauss-Kronrod rule from the matrix (1.16) determined by Laurie's algorithm. A Fortran implementation in quadruple precision arithmetic of Laurie's algorithm and the QR algorithm were used to compute highly accurate Gauss-Kronrod quadrature rules. The nodes and weights computed in quadruple precision were considered exact, and were used to determine the error in the quadrature rules computed by our and Laurie's algorithms in double precision arithmetic.

In our experiments we computed Gauss-Kronrod and Gauss-Kronrod-Radau quadrature rules associated with Jacobi measures

(5.1)
$$
dw(x) := c_0(1 - x)^\alpha (1 + x)^\beta dx, \qquad -1 < x < 1, \qquad \alpha, \beta > -1,
$$

where the scaling factor c_0, given by (4.8) with $a = -1$ and $b = 1$, is chosen to make $\mu_0 = 1$. Recursion coefficients for the associated orthogonal polynomials are explicitly known; see, e.g., [16].

TABLE 5.1. Errors in computed Gauss-Kronrod weights

n	α	β	d + c based alg.		mixed moment based alg.	
			max abs. error	max rel. error	max abs. error	max rel. error
10	-0.20	-0.99	8.68 E-16	4.42 E-15	1.29 E-14	2.36 E-13
10	-0.70	1.00	4.18 E-15	2.52 E-14	3.67 E-14	2.61 E-13
15	-0.97	-0.97	3.24 E-14	4.62 E-13	1.35 E-13	2.25 E-12
15	-0.99	-0.50	1.20 E-14	2.16 E-13	1.56 E-13	2.81 E-12
20	-0.60	-0.90	2.38 E-14	1.91 E-13	2.33 E-13	1.87 E-12
20	-0.99	-0.90	4.59 E-15	4.19 E-13	1.21 E-13	2.36 E-12

TABLE 5.2. Errors in computed Gauss-Kronrod nodes

n	α	β	d + c based alg.		mixed moment based alg.	
			max abs. error	max rel. error	max abs. error	max rel. error
10	-0.20	-0.99	5.86 E-16	4.65 E-15	2.50 E-15	2.50 E-15
10	-0.70	1.00	5.46 E-16	7.71 E-15	5.45 E-15	5.65 E-15
15	-0.97	-0.97	1.07 E-15	4.08 E-15	1.26 E-15	4.63 E-15
15	-0.99	-0.50	7.12 E-16	4.74 E-15	4.59 E-15	1.75 E-14
20	-0.60	-0.90	1.24 E-15	2.85 E-15	3.68 E-15	5.00 E-14
20	-0.99	-0.90	1.83 E-15	1.66 E-14	3.64 E-15	2.36 E-14

TABLE 5.3. Errors in computed Gauss-Kronrod-Radau weights, $\alpha = -0.99$, $\beta = -0.9$, fixed node at $x = -1$

n	d + c based alg.		mixed moment based alg.	
	max abs. error	max rel. error	max abs. error	max rel. error
9	3.46 E-15	6.35 E-14	8.12 E-14	1.46 E-12
15	1.62 E-14	2.67 E-13	8.52 E-14	1.58 E-12
21	1.42 E-14	6.22 E-13	2.86 E-13	5.40 E-12

TABLE 5.4. Errors in computed Gauss-Kronrod-Radau nodes, $\alpha = -0.99$, $\beta = -0.9$, fixed node at $x = -1$

n	d + c based alg.		mixed moment based alg.	
	max abs. error	max rel. error	max abs. error	max rel. error
9	4.62 E-16	3.31 E-15	2.74 E-15	5.64 E-15
15	9.89 E-16	1.30 E-14	2.95 E-15	8.63 E-15
21	2.27 E-15	1.08 E-14	4.76 E-15	4.76 E-15

TABLE 5.5. Errors in computed Gauss-Kronrod rules, $\alpha = -0.9999$, $\beta = -0.5$

n	d + c based alg.		mixed moment based alg.	
	max abs. error in weights	max abs. error in nodes	max abs. error in weights	max abs. error in nodes
16	7.87 E-16	9.84 E-16	2.11 E-15	2.64 E-15
32	3.52 E-15	1.07 E-15	1.28 E-14	1.15 E-15
64	1.64 E-15	1.77 E-15	5.50 E-14	4.10 E-15
128	3.80 E-14	2.18 E-15	2.12 E-13	2.43 E-15
256	8.28 E-14	1.52 E-15	2.36 E-12	5.53 E-15

Our computational results can be summarized as follows. For many choices of α and β in (5.1) both Laurie's and our methods yield high accuracy. However, when at least one of the exponents in (5.1) is fairly close to -1, our method generally gives smaller errors. The higher accuracy in the computed weights achieved by the method of the present paper is particularly noteworthy.

Tables 5.1 and 5.2 display the magnitudes of the largest absolute and relative errors in the computed Gauss-Kronrod nodes and weights. These errors are referred to as "max abs. error" and "max rel. error", respectively. We use the notation 5.11E-14 for $5.11 \cdot 10^{-14}$. The examples in the tables illustrate the performance of the methods for a variety of choices of α and β for a few fairly small values of n. When $\alpha = \beta$, the Gauss-Kronrod rule has a node at the origin by symmetry. In the example with $n = 15$ and $\alpha = \beta = -0.97$, we set the computed node closest to the origin to zero before computing absolute and relative errors of the nodes.

Tables 5.3 and 5.4 show the errors in a few computed Gauss-Kronrod-Radau rules associated with the measure (5.1) and a fixed node at $x = -1$. These rules were computed by applying Algorithm 1 to the measure (4.4). Due to the scaling ($\mu_0 = 1$) assumed by the algorithm, it follows from (4.8) that the weights determined by Algorithm 1 have to be scaled by the factor $s(\alpha, \beta) := 2B(1+\alpha, 2+\beta)/B(1+\alpha, 1+\beta)$ to yield the weights $\{\tilde{w}'_j\}_{j=1}^{2n+1}$ of the Gauss-Kronrod rule (4.5). These weights are required in (4.6) to determine the Gauss-Kronrod-Radau weights $\tilde{w}_{k,a}$. Table 5.3 shows the errors in the computed weights $\{\tilde{w}_{k,a}\}_{k=0}^{2n+1}$ for the Jacobi measure (5.1) with $\alpha = -0.99$ and $\beta = -0.9$. For these values of α and β, we have $s(\alpha, \beta) = 20/11$. Table 5.4 shows the error in the computed nodes $\{\tilde{x}_{k,a}\}_{k=0}^{2n+1}$. Finally, Table 5.5 illustrates the performance of the methods for some large values of n.

6. CONCLUSION

The paper describes a new algorithm for the· computation of Gauss-Kronrod quadrature rules and compares it to an algorithm recently proposed by Laurie. Both algorithms yield high accuracy for many problems. However, when an exponent in the Jacobi weight function (5.1) is close to -1, the algorithm of the present paper typically yields smaller errors. We also show how our algorithm can be applied to compute Kronrod extensions of Gauss-Radau and Gauss-Lobatto quadrature rules. The structure of the new algorithm makes efficient implementation in a parallel computing environment possible. This may be important in certain applications; see, e.g., [4, 8, 15].

ACKNOWLEDGMENTS

REFERENCES

[1] 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, 1992.

[2] D. Boley and G. H. Golub, A survey of matrix inverse eigenvalue problems, *Inverse Problems*, 3 (1987), pp. 595–622. MR **89m:**65036

[3] C. F. Borges and W. B. Gragg, A parallel divide and conquer algorithm for the generalized symmetric definite tridiagonal eigenvalue problem, *Numerical Linear Algebra*, (L. Reichel, A. Ruttan and R. S. Varga, eds.), de Gruyter, Berlin, 1993, pp. 11–29. MR **94k**:65051

[4] J. M. Bull and T. L. Freeman, Parallel globally adaptive quadrature on the KSR-1, *Adv. Comput. Math.*, 2 (1994), pp. 357–373. CMP 95:01

[5] D. Calvetti and L. Reichel, On an inverse eigenproblem for Jacobi matrices, *Adv. Comput. Math.*, 11 (1999), pp. 11–20.

[6] W. Gautschi, On generating orthogonal polynomials, *SIAM J. Sci. Stat. Comput.*, 3 (1982), pp. 289–317. MR **84e**:65022

[7] W. Gautschi, Gauss-Kronrod quadrature—a survey, *Numerical Methods and Approximation Theory* III, (G. V. Milovanović, ed.), University of Niš, 1987, pp. 39–66. MR **89k**:41035

[8] I. Gladwell, Vectorization of one dimensional quadrature codes, *Numerical Integration*, (P. Keast and G. Fairweather, eds.), Reidel, Dordrecht, 1987, pp. 231–238. MR **88i**:65039

[9] G. H. Golub and J. Kautsky, Calculation of Gauss quadratures with multiple free and fixed knots, *Numer. Math.*, 41 (1983), pp. 147–163. MR **84i**:65030

[10] G. H. Golub and J. H. Welsch, Calculation of Gauss quadrature rules, *Math. Comp.*, 23 (1969), pp. 221–230. MR **39**:6513

[11] M. Gu and S. C. Eisenstat, A divide-and-conquer algorithm for the symmetric tridiagonal eigenproblem, *SIAM J. Matrix Anal. Appl.*, 16 (1995), pp. 172–191. MR **95j**:65035

[12] D. P. Laurie, Calculation of Gauss-Kronrod quadrature rules, *Math. Comp.*, 66 (1997), pp. 1133–1145. MR **98m**:65030

[13] G. Monegato, A note on extended Gaussian quadrature rules, *Math. Comp.*, 30 (1976), pp. 812–817. MR **55**:13746

[14] G. Monegato, Stieltjes polynomials and related quadrature rules, *SIAM Rev.*, 24 (1982), pp. 137–158. MR **83d**:65067

[15] M. A. Napierala and I. Gladwell, Reducing ranking effects in parallel adaptive quadrature, in Proc. Seventh SIAM Conference on Parallel Processing for Scientific Computing, D. H. Bailey, J. R. Gilbert, M. V. Masagni, R. S. Schreiber, H. D. Simon, V. J. Torzon and L. T. Watson, eds., SIAM, Philadelphia, 1995, pp. 261–266.

[16] G. Szegő, Orthogonal Polynomials, 4th ed., Amer. Math. Society, Providence, 1975. MR **51**:8724

DEPARTMENT OF MATHEMATICS, CASE WESTERN RESERVE UNIVERSITY, CLEVELAND, OHIO 44106
E-mail address: dxc57@po.cwru.edu

DEPARTMENT OF COMPUTER SCIENCE, STANFORD UNIVERSITY, STANFORD, CALIFORNIA 94305
E-mail address: golub@chebyshev.stanford.edu

DEPARTMENT OF MATHEMATICS, NAVAL POSTGRADUATE SCHOOL, MONTEREY, CALIFORNIA 93943
E-mail address: gragg@nps.navy.mil

DEPARTMENT OF MATHEMATICS AND COMPUTER SCIENCE, KENT STATE UNIVERSITY, KENT, OHIO 44242
E-mail address: reichel@mcs.kent.edu