

## CALCULATION OF GAUSS-KRONROD QUADRATURE RULES

DIRK P. LAURIE

ABSTRACT. The Jacobi matrix of the  $(2n+1)$ -point Gauss-Kronrod quadrature rule for a given measure is calculated efficiently by a five-term recurrence relation. The algorithm uses only rational operations and is therefore also useful for obtaining the Jacobi-Kronrod matrix analytically. The nodes and weights can then be computed directly by standard software for Gaussian quadrature formulas.

### 1. INTRODUCTION

A  $(2n + 1)$ -point Gauss-Kronrod integration rule for the integral

$$(1) \quad If = \int_a^b f(x) \, ds(x),$$

where  $s$  is a nonnegative measure on the interval  $[a, b]$ , is a formula of the form

$$(2) \quad K^{(2n+1)}f = \sum_{i=1}^{2n+1} w_i f(x_i)$$

with the following two properties:

- $n$  of the nodes of  $K^{(2n+1)}$  coincide with those of the  $n$ -point Gaussian quadrature rule  $G^{(n)}$  for the same measure;
- $K^{(2n+1)}f = If$  whenever  $f$  is a polynomial of degree less than or equal to  $3n + 1$ .

A thorough survey of the history, existence and other theoretical properties, and computational aspects of Gauss-Kronrod rules and their generalizations is given by Gautschi [9]. In this paper we are concerned with the efficient calculation of the nodes  $x_i$  and weights  $w_i$  of Gauss-Kronrod rules. Several methods for computing these formulas have been suggested [2, 3, 4, 6, 14, 15, 21, 22] but most of them, as Gautschi puts it, compute the Gauss-Kronrod formula “piecemeal.” That is to say, the new points and their weights are found by one method, and the new weights for the old points by another.

The present author is aware of only two methods for computing the entire formula  $K^{(2n+1)}$  in a single algorithm. One of them [2, 6] is based on solving by Newton’s method the  $3n + 2$  equations that express the exactness of the quadrature rule. In [6] the authors remark about this method: “... by careful choice of initial approximations and continued monitoring of the iteration process, the method could be made to work for rules with up to 81 nodes ... .” The other [4] is a very general

---

Received by the editor June 28, 1995 and, in revised form, November 2, 1995.  
1991 *Mathematics Subject Classification*. Primary 65D30, 65D32.

algorithm for embedded quadrature formulas that uses among other tools the full eigenvector matrix of a tridiagonal matrix and the singular value decomposition. Both these methods therefore require  $O(n^3)$  operations.

The main contribution of the present paper is an  $O(n^2)$  procedure for computing specifically the Kronrod extension. It is not applicable to more general embedded quadratures. The procedure is derived in Section 4 and given in pseudocode in the Appendix. We do not claim that this procedure is more accurate than the “piecemeal” methods, or even that it is computationally more efficient — such issues will need a thorough investigation — but only that it is an efficient way of reducing the computation of a Kronrod rule to the well-studied problem of computing a Gaussian rule from recurrence coefficients.

For the calculation of Gaussian quadrature rules, the definitive algorithm (included in the recent software package of Gautschi [10]) is that of Golub and Welsch [11], which is based on the recurrence relation

$$(3) \quad p_{-1}(x) = 0;$$

$$(4) \quad p_0(x) = 1;$$

$$(5) \quad p_{j+1}(x) = (x - a_j)p_j(x) - b_j p_{j-1}(x), \quad j = 0, \dots, 1,$$

satisfied by the polynomials  $p_j$  orthogonal with respect to the weight  $s$ . Since  $b_0$  only appears as a multiplier for  $p_{-1}$  (which is zero), any finite value will do: we follow Gautschi [7] in putting  $b_0 = Ip_0^2$ , which leads to the useful property that

$$(6) \quad Ip_j^2 = b_0 b_1 \dots b_j.$$

Golub and Welsch show that for all integers  $m \geq 1$  the nodes of the Gaussian formula  $G^{(m)}$  are the eigenvalues, and the weights are proportional to the squares of the first components of the normalized eigenvectors, of the symmetric tridiagonal matrix (known as the Jacobi matrix associated with the Gaussian quadrature formula)

$$(7) \quad T_m = \begin{bmatrix} a_0 & \sqrt{b_1} & & & \\ \sqrt{b_1} & a_1 & \sqrt{b_2} & & \\ & \ddots & \ddots & \ddots & \\ & & \sqrt{b_{m-1}} & a_{m-1} & \\ & & & & \end{bmatrix}.$$

*Remark.* Although the routine in [10] works explicitly with the quantities in  $T_m$  (which are formed by taking square roots of the  $b_j$ ) there exist square-root free eigensolvers for tridiagonal matrices that work directly with the  $b_j$  (see [19], Section 8–15, and [20]). An investigation of which eigensolver is most accurate for the special purpose of computing Gaussian quadrature formulas would be worthwhile, but is outside the scope of the present article. The claim in [20] that small components of eigenvectors are computed with high relative accuracy is of particular interest here.

Let us now suppose that the  $2n + 1$  nodes and weights of the Gauss-Kronrod formula  $K^{(2n+1)}$  can be found in a similar way from the symmetric tridiagonal

matrix

$$(8) \quad \widehat{T}_{2n+1} = \begin{bmatrix} \widehat{a}_0 & \sqrt{\widehat{b}_1} & & & \\ \sqrt{\widehat{b}_1} & \widehat{a}_1 & \sqrt{\widehat{b}_2} & & \\ & & \ddots & \ddots & \\ & & & \sqrt{\widehat{b}_{2n}} & \widehat{a}_{2n} \end{bmatrix}$$

which we shall call the Jacobi-Kronrod matrix associated with the Gauss-Kronrod formula  $K^{(2n+1)}$ . We do not know  $\widehat{T}_{2n+1}$ , but we do know a lot about it: some theoretical questions are discussed in Section 2. Our main result in Section 4 is a rational algorithm that computes  $\widehat{T}_{2n+1}$  efficiently in  $O(n^2)$  arithmetic operations.

A related problem is considered by Boley and Golub [1], where the double-dimension Jacobi matrix  $T_{2n}$  is to be found when all its eigenvalues are known and  $T_n$  is specified. They use the available information to compute the weights of  $G^{(2n)}$  in  $O(n^2)$  arithmetic operations, after which any algorithm (three such are cited in [1]) for recovering a Jacobi matrix from its Gaussian quadrature formula may be used to compute  $T_{2n}$ . Since we do not know all the eigenvalues of  $\widehat{T}_{2n+1}$ , a similar algorithm is not possible here.

The main tool that we require is the theory of mixed moments, which is implicit in the work of Salzer [25] and appears more explicitly in the work of Sack and Donovan [24] and Wheeler [26]. An accessible exposition is given by Gautschi [8]. For the sake of clarity, we give the essentials of the theory in Section 3.

## 2. PROPERTIES OF THE JACOBI-KRONROD MATRIX

In the literature on Kronrod formulas and Stieltjes polynomials (see the surveys [9, 18]) some non-existence theorems on these formulas are given. It is therefore of interest to relate existence questions to the Jacobi-Kronrod matrix. We use the following terminology [16]:

- A quadrature formula *exists* if its defining equations have a (possibly complex) solution.
- The formula is *real* if the points and weights are all real.
- A real formula is *internal* if all the points belong to the (closed) interval of integration. A node not belonging to the interval is called an *exterior node*.
- The formula is *positive* if all the weights are positive.

It is well known (see e.g. [7]) that there is a one-to-one correspondence between Jacobi matrices and quadrature formulae with positive weights: if we knew the Kronrod formula itself, we could in principle find the Jacobi-Kronrod matrix, even though the computation may be delicate [12]. So we have:

**Fact 1.** The Jacobi-Kronrod matrix exists and is real if and only if the corresponding Kronrod formula exists and is real and positive.

Note that this fact does not imply that the Kronrod formula contains no exterior nodes. In view of Monegato's result [17] that positivity of the weights of the new points is equivalent to the interlacing property (i.e. that there is one Gaussian node between any consecutive pair of Kronrod nodes), all that can be said is that at most two nodes, one at each end point, are exterior. It is, however, easy to diagnose whether the formula is interior once the Jacobi-Kronrod matrix is available:

one simply factorizes  $\widehat{T}_{2n+1} - aI = LDL^T$ , where  $L$  is lower triangular with unit diagonal, and  $D = \text{diag}\{d_0, d_1, \dots, d_{2n}\}$ . It is well known (see any standard text on linear algebra) that there is one eigenvalue of  $\widehat{T}_{2n+1}$  less than the end point  $a$  if and only if there is one sign change on the diagonal of  $D$ . This algorithm is also rational, since  $d_0 = a_0$ ,  $d_{j+1} = a_{j+1} - b_{j+1}/d_j$ . A similar test can be applied at the other end point  $b$ .

As pointed out in [16], the fact that the formula  $K^{(2n+1)}$  is exact for polynomials of degree less than or equal to  $3n + 1$  implies that the first  $3n + 1$  coefficients in the sequence  $\{\widehat{a}_0, \widehat{b}_1, \widehat{a}_1, \widehat{b}_2, \dots\}$  equal the corresponding coefficients in the known sequence  $\{a_0, b_1, a_1, b_2, \dots\}$ . Only the remaining  $n$  coefficients are unknown. They are determined by the condition that  $n$  of the eigenvalues of  $\widehat{T}_{2n+1}$  are fixed to be equal to the eigenvalues of  $T_n$ . At first sight this seems to be a partial inverse eigenvalue problem, subject to all the difficulties that surround the solution of such problems, but we shall show in Section 4 that the  $n$  unknown coefficients can be determined efficiently in  $O(n^2)$  arithmetic operations.

The algorithm given in Section 4 is rational and the only divisions are by quantities that can only be zero if some  $\widehat{b}_j = 0$ . We therefore have:

**Fact 2.** The algorithm of Section 4 cannot break down if the corresponding Kronrod formula exists and is real and positive.

This does not imply that the algorithm will break down if the Gauss-Kronrod formula is not real and positive: it may very well still succeed. But in that case, by Fact 1, the Kronrod-Jacobi matrix cannot be real, so at least one of the computed recurrence coefficients  $\widehat{b}_j$  must be negative. This means, of course, that the Golub-Welsch algorithm, which always produces real non-negative formulas, can no longer be used to compute the Gauss-Kronrod formula.

There does not seem to be an easy way of distinguishing between the cases of negative weights and non-real nodes. For example, when trying to construct a 7-point extension of the 3-point Gauss-Hermite formula (proved to be impossible in [13]), we obtain  $\widehat{b}_6 = -1$ , and indeed the corresponding Kronrod formula has complex nodes. When trying to construct a 9-point extension of the 4-point Gauss-Hermite formula (not proved to be impossible in [13]<sup>1</sup>) we obtain  $\widehat{b}_7 = -\frac{1}{4}$ ,  $\widehat{b}_8 = \frac{1}{4}$ , and the formula indeed turns out to have real nodes but negative weights at two of the old points.

It is also possible to use the Jacobi-Kronrod matrix as a theoretical tool. In [5] the authors investigate whether the Gauss-Kronrod formula associated with the Jacobi weight function

$$(9) \quad ds(x) = (1-x)^\alpha(1+x)^\beta dx$$

is real, positive and internal. In the case  $n = 1$  they obtain analytic results, and thereafter give graphs based on numerical calculations. It is easy to code the algorithm of Section 4 in an algebraic language and obtain the numbers  $\widehat{b}_j$  and  $d_j$  analytically. To illustrate this point we have computed the critical coefficients in

<sup>1</sup>The results of [13] are often misquoted as implying that a positive extension of the 4-point Hermite formula exists, but the abstract only claims "... do not exist with positive weights when  $n > 0$  in the Laguerre case and  $n = 3$  or  $n > 4$  in the Hermite case." The corollary on p. 985 of [13] does state "... only exist for  $n = 1, 2, 4$ " but the proof is a non-existence proof of the other cases, so the quoted phrase is clearly a misprint for "... can only exist for  $n = 1, 2, 4$ ."

the case  $n = 2$ , namely

$$\widehat{b}_4 = \frac{2(\alpha + \beta + 5)P_{3,3}(\alpha, \beta)}{(4 + \beta)(\alpha + \beta + 3)(\alpha + \beta + 4)(\alpha + \beta + 6)^2},$$

$$d_4 = \frac{P_{2,4}(\alpha, \beta)}{(\alpha + \beta + 3)(\alpha + \beta + 6)^2(\alpha + \beta + 8)^2},$$

where

$$P_{2,4}(\alpha, \beta) = [ 1 \quad \alpha \quad \alpha^2 ] \begin{bmatrix} 72 & 213 & 137 & 21 & 1 \\ -3 & 4 & 21 & 2 & 0 \\ 11 & 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ \beta \\ \beta^2 \\ \beta^3 \\ \beta^4 \end{bmatrix}$$

and

$$P_{3,3}(\alpha, \beta) = [ 1 \quad \alpha \quad \alpha^2 \quad \alpha^3 ] \begin{bmatrix} 576 & 336 & -71 & -23 \\ 336 & 434 & 51 & 1 \\ -71 & 51 & 2 & 0 \\ -23 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ \beta \\ \beta^2 \\ \beta^3 \end{bmatrix}.$$

One can therefore conclude that the 5-point Gauss-Kronrod formula is real and positive when  $P_{3,3}(\alpha, \beta) > 0$  and is internal when  $P_{2,4}(\alpha, \beta) \geq 0$  (there is no node less than  $-1$ ) and  $P_{2,4}(\beta, \alpha) \geq 0$  (there is no node greater than  $1$ ). The lines marked  $b$  and  $c$  in the first graph on p.241 of [5] are thereby obtained analytically.

The following lemma gives an essential property of the Jacobi-Kronrod matrix  $\widehat{T}_{2n+1}$ , which will later be the key to its efficient computation.

**Lemma 1.** *The characteristic polynomial of the trailing principal  $n \times n$  submatrix of the Jacobi-Kronrod matrix  $\widehat{T}_{2n+1}$  is the same as that of its leading principal  $n \times n$  submatrix.*

*Proof.* Denote by  $\phi_k$  and  $\psi_k$  respectively the characteristic polynomial of the leading and trailing  $k \times k$  principal submatrices of  $\widehat{T}_{2n+1}$ . Expand  $\phi_{2n+1}(\lambda) = \det(\widehat{T}_{2n+1} - \lambda I)$  along the  $(n + 1)$ -st row. Then (suppressing the argument  $(\lambda)$  for the sake of readability)

$$(10) \quad \phi_{2n+1} = -\phi_{n-1}\widehat{b}_n\psi_n + (a_n - \lambda)\phi_n\psi_n - \phi_n\widehat{b}_{n+1}\psi_{n-1}.$$

Clearly any common zero of  $\phi_n$  and  $\psi_n$  is a zero of  $\phi_{2n+1}$ . Conversely, if  $\phi_{2n+1}$  is to have  $\phi_n$  as a factor, then  $\phi_{n-1}\psi_n$  must be divisible by  $\phi_n$  since  $\widehat{b}_n = b_n$ , which is nonzero by (6). But  $\phi_{n-1}$  and  $\phi_n$  are consecutive terms in a sequence of orthogonal polynomials and therefore mutually prime. It follows that  $\psi_n$  is divisible by  $\phi_n$ . Since the two polynomials have the same leading coefficient, they must be identical.  $\square$

*Remark.* Once we know that  $\phi_n = \psi_n$ , it is of course possible to divide equation (10) by  $\psi_n$  to obtain

$$(11) \quad \frac{\phi_{2n+1}}{\psi_n} = -\phi_{n-1}\widehat{b}_n + (a_n - \lambda)\phi_n - \widehat{b}_{n+1}\psi_{n-1},$$

which gives an explicit expression for the polynomial with zeros at the Kronrod nodes.

## 3. MIXED MOMENTS

Suppose that two different systems of monic orthogonal polynomials are being considered: the system defined in (5), which we shall sometimes call the ‘old’ system, and a similar system (the ‘new’ system) defined by

$$(12) \quad \pi_{-1}(x) = 0;$$

$$(13) \quad \pi_0(x) = 1;$$

$$(14) \quad \pi_{j+1}(x) = (x - \alpha_j)\pi_j(x) - \beta_j\pi_{j-1}(x), \quad j = 0, 1, \dots$$

The new system (12–14) is assumed to be orthogonal with respect to some inner product  $(\cdot, \cdot)_\sigma$  with the property that for any three polynomials  $f, g, h$ ,

$$(15) \quad (f, gh)_\sigma = (fg, h)_\sigma.$$

As before, it is convenient to put  $\beta_0 = (\pi_0, \pi_0)_\sigma$ .

Salzer [25] considered the problem of converting a finite expansion in terms of the old polynomials  $p_j$  into one in terms of the new polynomials  $\pi_j$ . Expanding each  $p_j$  in terms of the  $\pi_j$ , one finds

$$(16) \quad p_l = \sum_{k=0}^l \frac{\sigma_{k,l}}{\sigma_{k,k}} \pi_k$$

where the mixed moments  $\sigma_{k,l}$  are given by

$$(17) \quad \sigma_{k,l} = (\pi_k, p_l)_\sigma.$$

The crucial observation is that a five-term recurrence for the mixed moments can be derived by putting  $j = l$  in (5),  $j = k$  in (14), taking the new inner product with  $\pi_k$  and  $p_l$  respectively, and subtracting. Thanks to the property (15), the terms that do not immediately reduce to mixed moments cancel, and we are left with

$$(18) \quad \sigma_{k,l+1} + a_l\sigma_{k,l} + b_l\sigma_{k,l-1} - (\sigma_{k+1,l} + \alpha_k\sigma_{k,l} + \beta_k\sigma_{k-1,l}) = 0.$$

The equation (18) can be written in a visually appealing manner familiar to practitioners of finite difference methods as

$$(19) \quad \left\{ \begin{array}{ccc} & -\beta_k & \\ b_l & a_l - \alpha_k & 1 \\ & -1 & \end{array} \right\} \sigma_{k,l} = 0.$$

Here  $k$  is the row index and  $l$  is the column index, and the rows are numbered from top to bottom as is usual with matrices. Values of  $\sigma_{k,l}$  that are known in advance are

$$(20) \quad \sigma_{0,0} = \beta_0;$$

$$(21) \quad \sigma_{-1,l} = 0, \quad l = 0, 1, \dots, n,$$

$$(22) \quad \sigma_{k,-1} = 0, \quad k = 0, 1, \dots, n,$$

$$(23) \quad \sigma_{k,l} = 0, \quad l = 0, 1, \dots, k-1.$$

Equation (23) holds because  $\pi_k$  is orthogonal in the new inner product to all polynomials of lower degree.

In the application considered by Salzer [25], all the new recurrence coefficients are known. Then  $\sigma_{k,l}$  can be computed for  $l$  increasing, with the ‘East’ moment in (19) the unknown, as in Figure 1. The solution to the problem is obtained by substituting (16) into the given orthogonal expansion. (This explanation of

|     |     |           |   |       |          |
|-----|-----|-----------|---|-------|----------|
|     | $l$ | 0         | 1 | 2     | 3        |
| $k$ |     | 0         | 0 | 0     | 0        |
| 0   | 0   | $\beta_0$ | * | $\nu$ | $\dots$  |
| 1   | 0   | 0         | * | $\nu$ | $\dots$  |
| 2   |     | 0         | 0 | $\nu$ | $\dots$  |
| 3   |     |           | 0 | 0     | $\ddots$ |
| 4   |     |           |   | 0     | 0        |

FIGURE 1. Build-up of mixed moments  $\sigma_{k,l}$  in Salzer’s algorithm. The stars denote moments that have already been computed, and each  $\nu$  denotes a new moment that is being computed at that stage.

|     |     |         |         |         |          |         |              |              |              |              |
|-----|-----|---------|---------|---------|----------|---------|--------------|--------------|--------------|--------------|
|     | $l$ | 0       | 1       | 2       | 3        | $\dots$ | $2n - 4$     | $2n - 3$     | $2n - 2$     | $2n - 1$     |
| $k$ |     | 0       | 0       | 0       | 0        | $\dots$ | 0            | 0            | 0            | 0            |
| 0   | 0   | $\mu_0$ | $\mu_1$ | $\mu_2$ | $\mu_3$  | $\dots$ | $\mu_{2n-4}$ | $\mu_{2n-3}$ | $\mu_{2n-2}$ | $\mu_{2n-1}$ |
| 1   | 0   | 0       | *       | *       | *        | $\dots$ | *            | *            | *            |              |
| 2   |     | 0       | 0       | $\nu$   | $\nu$    | $\dots$ | $\nu$        | $\nu$        |              |              |
| 3   |     |         | 0       | 0       | $\ddots$ | $\dots$ | $\ddots$     |              |              |              |

FIGURE 2. Build-up of mixed moments in the Sack-Donovan-Wheeler algorithm. The stars denote moments that have already been computed, and each  $\nu$  denotes a new moment that is being computed at that stage.

Salzer’s algorithm, simplified to emphasize its kinship with the other algorithms considered here, does not do full justice to its ingenuity. In fact, it employs a five-term recurrence involving the expansion coefficients directly, rather than doing the last-mentioned substitution.)

In the application considered by Sack and Donovan [24] and Wheeler [26], the new recurrence coefficients  $\alpha_k$  and  $\beta_k$  are unknown, but the modified moments

$$(24) \quad \mu_l = \sigma_{0,l}, \quad l = 0 = 1, \dots, 2n - 1$$

are assumed to be known. In that case, the recurrence can be computed for  $k$  increasing, with the ‘South’ moment in (19) the unknown, as in Figure 2. The essential idea is that when row  $k$  has been computed, one can obtain  $\beta_k$  and  $\alpha_k$  by considering (19) at position  $(k, k - 1)$  to yield

$$(25) \quad \beta_k = \frac{\sigma_{k,k}}{\sigma_{k-1,k-1}}$$

and at position  $(k, k)$  to yield

$$(26) \quad \alpha_k = a_k + \frac{\sigma_{k,k+1} - \beta_k \sigma_{k-1,k}}{\sigma_{k,k}}$$

$$(27) \quad = a_k + \frac{\sigma_{k,k+1}}{\sigma_{k,k}} - \frac{\sigma_{k-1,k}}{\sigma_{k-1,k-1}}.$$

#### 4. COMPUTATION OF THE JACOBI-KRONROD MATRIX

The lemma in Section 2 reduces the partial inverse eigenvalue problem of finding a  $(2n + 1) \times (2n + 1)$  matrix with  $n$  prescribed eigenvalues to a standard inverse eigenvalue problem in which the number of prescribed eigenvalues is the same as the order of the matrix. Moreover, this problem can be solved in a finite number of steps.

Let the old polynomials  $p_l$  of §2 be the same as those of §1, and let the new polynomials  $\pi_k$  of §2 be given by (12–14) with

$$(28) \quad \alpha_k = \widehat{a}_{n+k+1}, \beta_k = \widehat{b}_{n+k+1}, k = 0, 1, \dots, n - 1.$$

That is, the polynomials  $\pi_k$  are those associated with the trailing  $n \times n$  submatrix of the Jacobi-Kronrod matrix.

Which mixed moments can we now compute? The first  $n - 1$  entries in the sequence  $\{\alpha_0, \beta_1, \alpha_1, \beta_2, \dots\}$  are known: they are just  $a_{n+1}, b_{n+2}, a_{n+2}, b_{n+3}, \dots$ . So we can compute a number of mixed moments with the East moment as the unknown, as in the case of Salzer's algorithm. To progress further, we use the fact that  $\pi_n = p_n$ , and therefore

$$(29) \quad \sigma_{k,n} = 0, k = 0, 1, \dots, n - 1.$$

This allows us to complete the process in the same way as the Sack-Donovan-Wheeler algorithm, with the South moment as the unknown, computing new values of  $\alpha_k$  and  $\beta_k$  as soon as possible, up to  $\sigma_{n-1,n-1}$ , which together with the known  $\sigma_{n-1,n} = 0$  is all that we need. The whole scheme is illustrated for  $n = 7$  in Figure 3. Here we know  $\alpha_0, \beta_1, \alpha_1, \beta_2, \alpha_2$  and  $\beta_3$ . The pattern is typical for odd  $n$ ; when  $n$  is even, a diagonal of asterisked entries starting at  $\sigma_{k,k+1}$  where  $k = n/2 - 1$  can also be computed, because  $a_k$  is then known.

| $k$ | $l$ | 0 | 1 | 2 | 3 | 4     | 5     | 6     | $n$ |
|-----|-----|---|---|---|---|-------|-------|-------|-----|
|     |     | 0 | 0 | 0 | 0 | 0     | 0     | 0     | 0   |
| 0   | 0   | * | * | * | * | *     | *     | *     | 0   |
| 1   | 0   | 0 | * | * | * | *     | *     | $\nu$ | 0   |
| 2   |     | 0 | 0 | * | * | *     | $\nu$ | $\nu$ | 0   |
| 3   |     |   | 0 | 0 | * | $\nu$ | $\nu$ | $\nu$ | 0   |
| 4   |     |   |   | 0 | 0 | $\nu$ | $\nu$ | $\nu$ | 0   |
| 5   |     |   |   |   | 0 | 0     | $\nu$ | $\nu$ | 0   |
| 6   |     |   |   |   |   | 0     | 0     | $\nu$ | 0   |

FIGURE 3. Computation of mixed moments for the Jacobi-Kronrod matrix when  $n = 7$ . The stars denote quantities computable from the known recurrence coefficients, with the recurrence going 'Eastwards', and the  $\nu$ 's quantities computable with the recurrence going 'Southwards', by using the fact that  $p_n = \pi_n$  gives a column of zeros, here indicated by 0.

Building the table diagonal by diagonal, we obtain the following algorithm.

*Initialization:*

$$\begin{aligned}\sigma_{0,0} &= 1; \\ \sigma_{-1,l} &= 0, \quad l = 0 : n; \\ \sigma_{k,n} &= 0, \quad k = 1 : n - 1; \\ \sigma_{k,l} &= 0, \quad l = 1 : n - 2, \quad k = l + 1, l + 2.\end{aligned}$$

*Eastward phase:* For  $m = 0 : n - 2$

$$\begin{aligned}\sigma_{k,l+1} &= \sigma_{k+1,l} + (\alpha_k - a_l)\sigma_{k,l} + \beta_k\sigma_{k-1,l} - b_l\sigma_{k,l-1}, \\ k &= \lceil m/2 \rceil : -1 : 0, \quad l = m - k.\end{aligned}$$

*Southward phase:* For  $m = n - 1 : 2n - 3$

$$\begin{aligned}\sigma_{k+1,l} &= \sigma_{k,l+1} - (\alpha_k - a_l)\sigma_{k,l} - \beta_k\sigma_{k-1,l} + b_l\sigma_{k,l-1}, \\ k &= m + 1 - n : \lceil m/2 \rceil, \quad l = m - k.\end{aligned}$$

$$\text{If } m \text{ is even, } \alpha_k = a_k + \frac{\sigma_{k,k+1} - \beta_k\sigma_{k1,k}}{\sigma_{k,k}}.$$

$$\text{If } m \text{ is odd, } \beta_k = \frac{\sigma_{k,k}}{\sigma_{k-1,k-1}}.$$

*Termination:*

$$\alpha_{n-1} = a_{n-1} - \beta_{n-1} \frac{\sigma_{n-2,n-1}}{\sigma_{n-1,n-1}}.$$

The algorithm uses approximately  $\frac{3}{2}n^2$  multiplications, and can be implemented with two temporary vectors of length  $\lceil n/2 \rceil + 2$ . The details are given in the Appendix.

*Remark.* A variation may be obtained by reversing the rows and columns of the leading  $n \times n$  submatrix of  $\widehat{T}_{2n+1}$  before entering the algorithm. Since the characteristic polynomial is unchanged, the values of  $\alpha_k$  and  $\beta_k$  computed in this way are theoretically the same, even though the modified moments  $\sigma_{k,j}$  differ. Our numerical experiments do not indicate a preference for either version.

## 5. NUMERICAL EXAMPLE

We illustrate the algorithm by computing the Kronrod extension of the  $n$ -point Gauss-Jacobi formula for the weight function (9) over the interval  $(-1, 1)$ . In the numerical example shown here, we took  $\alpha = 0.3$  and  $\beta = -0.6$ , and  $n = 3, 4, \dots, 199$ . The computations were performed in double precision on a machine conforming to the IEEE floating point standard, and for both the normal and ‘reversed’ versions of the algorithm. Actually, as recommended by Reichel [23], the coefficients were scaled to the interval  $(-2, 2)$  before performing the computation, and the resulting points and weights (obtained by Gautschi’s routine **gauss** from [10]) were scaled back to  $(-1, 1)$ . On a machine with a small exponent range, this subterfuge is necessary to avoid underflow of the modified moments.

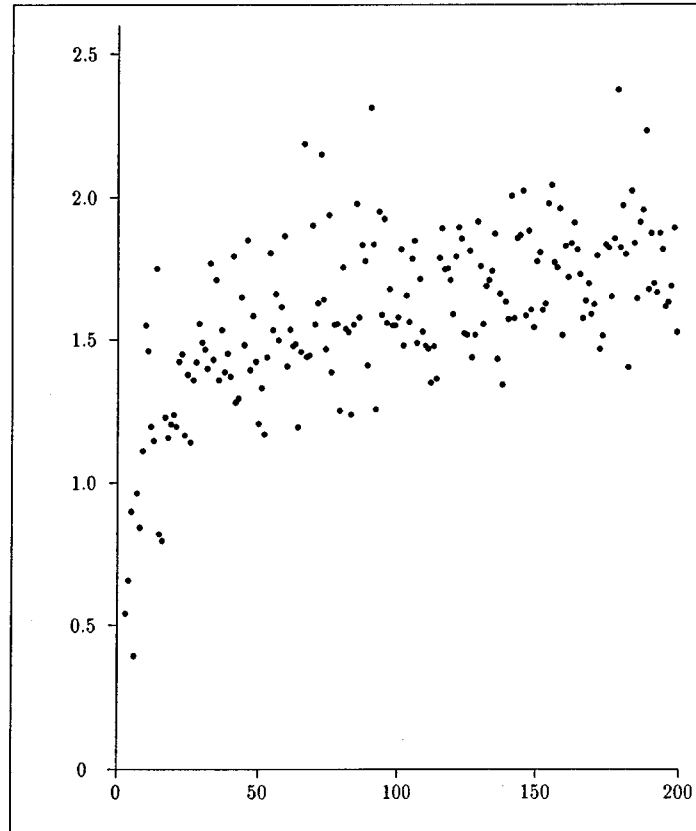


FIGURE 4. Mean absolute difference (units of machine epsilon) in common points of computed  $n$ -point Gauss-Jacobi and  $(2n + 1)$ -point Kronrod-Jacobi formulas ( $\alpha = 0.3, \beta = -0.6$ ).

In this example the coefficients  $\hat{b}_j$  all turn out to be positive, and therefore the old and new nodes interlace. There is one node less than  $-1$ . An *a posteriori* indication of the accuracy of the computed formula can be obtained by comparing every second computed node of the Kronrod extension with the corresponding node  $\xi_i$  of the original Gaussian formula, computed using **gauss** in the same precision. Since there is no reason to think that the eigenvalue solver can discriminate in accuracy between the original and the new nodes, one may suppose that the new nodes are computed to an accuracy similar to that observed for the old nodes. Unfortunately this check says nothing about the accuracy of the weights.

In Figure 4 we show the average difference between these zeros as a multiple of the machine roundoff level  $\tau$ , i.e. the quantity

$$\eta_n = \frac{1}{n} \sum_{i=1}^n \frac{|x_{2i} - \xi_i|}{\tau}.$$

Most of the values lie in the range  $1 < \eta_n < 2$ , although there is a slight tendency

for  $\eta_n$  to increase with  $n$ . The results from the ‘reversed’ version show essentially the same behaviour and are therefore not shown.

**Thanks.** The suggestions of an anonymous referee have greatly improved this paper.

#### APPENDIX A. PSEUDOCODE FOR THE ALGORITHM

*Input:* Vectors  $\mathbf{a} = \{a_0, a_1, \dots, a_{2n}\}$  and  $\mathbf{b} = \{b_0, b_1, \dots, b_{2n}\}$ , in which the elements  $a_0, a_1, \dots, a_{\lfloor 3n/2 \rfloor}$  and  $b_0, b_1, \dots, b_{\lfloor 3n/2 \rfloor}$  have been initialized to the recurrence coefficients of the given weight.

*Working storage:* Vectors  $\mathbf{s} = \{s_{-1}, s_0, \dots, s_{\lfloor n/2 \rfloor}\}$  and  $\mathbf{t} = \{t_{-1}, t_0, \dots, t_{\lfloor n/2 \rfloor}\}$ , initialized to zeros.

*Output:* The vectors  $\mathbf{a}$  and  $\mathbf{b}$  now contain the recurrence coefficients from which the  $(2n+1)$ -point Gauss-Kronrod formula is obtained by Gautschi’s routine **gauss** from [10].

We use the notation  $\mathbf{s} \leftrightarrow \mathbf{t}$  to indicate that the contents of the vectors  $\mathbf{s}$  and  $\mathbf{t}$  should be swapped. In an actual implementation, only the pointers to the vectors would be swapped: e.g. the vectors could be rows  $i_s$  and  $i_t$  of a two-row matrix, and one would swap the indices  $i_s$  and  $i_t$ .

```

begin generate Jacobi-Kronrod matrix
   $t_0 \leftarrow b_{n+1}$ 
  for  $m$  from 0 to  $n-2$ 
     $u \leftarrow 0$ 
    for  $k$  from  $\lfloor (m+1)/2 \rfloor$  downto 0
       $l \leftarrow m-k$ 
       $u \leftarrow u + (a_{k+n+1} - a_l)t_k + b_{k+n+1}s_{k-1} - b_l s_k$ 
       $s_k \leftarrow u$ 
    end for  $k$ 
     $\mathbf{s} \leftrightarrow \mathbf{t}$ 
  end for  $m$ 
  for  $j$  from  $\lfloor n/2 \rfloor$  downto 0
     $s_j \leftarrow s_{j-1}$ 
  end for  $j$ 
  for  $m$  from  $n-1$  to  $2n-3$ 
     $u \leftarrow 0$ 
    for  $k$  from  $m+1-n$  to  $\lfloor (m-1)/2 \rfloor$ 
       $l \leftarrow m-k$ 
       $j \leftarrow n-1-l$ 
       $u \leftarrow u - (a_{k+n+1} - a_l)t_j - b_{k+n+1}s_j + b_l s_{j+1}$ 
       $s_j \leftarrow u$ 
    end for  $k$ 
    if  $m \bmod 2 = 0$ 
       $k \leftarrow m/2$ 
       $a_{k+n+1} \leftarrow a_k + (s_j - b_{k+n+1}s_{j+1})/t_{j+1}$ 
    else
       $k \leftarrow (m+1)/2$ 
       $b_{k+n+1} \leftarrow s_j/s_{j+1}$ 

```

```

endif
s ↔ t
end for m
a2n ← an-1 - b2ns0/t0
end generate Jacobi-Kronrod matrix

```

## REFERENCES

1. D. Boley and G. H. Golub. A survey of matrix inverse eigenvalue problems. *Inverse problems*, 3:595–622, 1987. MR **89m**:65036
2. F. Caliò, W. Gautschi, and E. Marchetti. On computing Gauss-Kronrod quadrature formulae. *Math. Comput.*, 47:639–650, 1986. MR **88a**:65028
3. F. Caliò and E. Marchetti. Derivation and implementation of an algorithm for singular integrals. *Computing*, 38:235–245, 1987. MR **88h**:65048
4. S. Elhay and J. Kautsky. Generalized Kronrod Patterson type embedded quadratures. *Aplikace Matematiky*, 37:81–103, 1992. MR **92j**:65026
5. W. Gautschi and S. Notaris. An algebraic study of Gauss-Kronrod quadrature formulae for Jacobi weight functions. *Math. Comput.*, 51: 231–248, 1988. MR **89f**:65031
6. W. Gautschi and S. Notaris. Newton's method and Gauss-Kronrod quadrature. In H. Braß and G. Hämmerlin, editors, *Numerical Integration III*, pages 60–71, Basel, 1988. Birkhäuser. MR **90m**:65104
7. Walter Gautschi. On generating orthogonal polynomials. *SIAM J. Scient. Statist. Comput.*, 3:289–317, 1982. MR **84e**:65022
8. Walter Gautschi. Questions of numerical condition related to polynomials. In Gene H. Golub, editor, *Studies in Numerical Analysis*, pages 140–177. Math. Assoc. Amer., 1984. CMP 20:07
9. Walter Gautschi. Gauss-Kronrod quadrature — a survey. In G. V. Milovanović, editor, *Numerical Methods and Approximation Theory III*, pages 39–66. University of Niš, 1987. MR **89k**:41035
10. Walter Gautschi. Algorithm 726: ORTHPOL — a package of routines for generating orthogonal polynomials and Gauss-type quadrature rules. *ACM Trans. Math. Software*, 20:21–62, 1994.
11. G.H. Golub and J.H. Welsch. Calculation of Gauss quadrature rules. *Math. Comput.*, 23:221–230, 1969. MR **39**:6513
12. W. B. Gragg and W. J. Harrod. The numerically stable reconstruction of Jacobi matrices from spectral data. *Numer. Math.*, 44:317–335, 1984. MR **85i**:65052
13. D. K. Kahaner and G. Monegato. Nonexistence of extended Gauss-Laguerre and Gauss-Hermite quadrature rules with positive weights. *Z. Angew. Math. Phys.*, 29:983–986, 1978. MR **80d**:65034
14. J. Kautsky and S. Elhay. Gauss quadratures and Jacobi matrices for weight functions not of one sign. *Math. Comput.*, 43:543–550, 1984. MR **86f**:65049
15. A. S. Kronrod. *Nodes and Weights of Quadrature Formulas*. Consultants Bureau, New York, 1965. MR **32**:598
16. D. P. Laurie. Anti-Gaussian quadrature formulas. *Math. Comput.*, 65: 739–747, 1996. MR **96m**:65026
17. G. Monegato. A note on extended Gaussian quadrature rules. *Math. Comput.*, 30:812–817, 1976. MR **55**:13746
18. G. Monegato. Stieltjes polynomials and related quadrature rules. *SIAM Review*, 24:137–158, 1982. MR **83d**:65067
19. Beresford N. Parlett. *The Symmetric Eigenvalue Problem*. Prentice-Hall, Englewood Cliffs, 1980. MR **81j**:65063
20. Beresford N. Parlett. The new qd algorithms. In *Acta Numerica 1995*, pages 459–491. Cambridge University Press, 1995. MR **96g**:65041
21. T. N. L. Patterson. The optimum addition of points to quadrature formulae. *Math. Comput.*, 22:847–856, 1968. MR **39**:3701
22. R. Piessens and M. Branders. A note on the optimal addition of abscissas to quadrature formulas of Gauss and Lobatto type. *Math. Comput.*, 28:135–139 and 344–347, 1974. MR **49**:8293
23. L. Reichel. Newton interpolation at Leja points. *BIT*, 30:332–346, 1990. MR **91e**:65020

24. R.A. Sack and A.F. Donovan. An algorithm for Gaussian quadrature given modified moments. *Numer. Math.*, 18:465–478, 1972. MR **46**:2829
25. H. E. Salzer. A recurrence scheme for converting from one orthogonal expansion into another. *Comm. ACM.*, 16:705–707, 1973. MR **52**:15956
26. J.C. Wheeler. Modified moments and Gaussian quadratures. *Rocky Mountain J. Math.*, 4:287–296, 1974. MR **48**:12785

POTCHEFSTROOM UNIVERSITY FOR CHRISTIAN HIGHER EDUCATION, P. O. BOX 1174, VANDERBILJPARK, 1900, SOUTH AFRICA

*E-mail address:* `dlaurie@na-net.ornl.gov`