Efficient High-Accuracy Quadrature for Two Classes of Logarithmic Weight Functions

JAMES S. BALL
and
NELSON H. F. BEEBE
University of Utah

Integrals with logarithmic singularities are often difficult to evaluate by numerical methods. In this work, a quadrature method is developed that allows the exact evaluation (up to machine accuracy) of integrals of polynomials with two general types of logarithmic weights.

The total work for the determination of N nodes and points of the quadrature method is $\mathcal{O}(N^2)$. Subsequently, integrals can be evaluated with $\mathcal{O}(N)$ operations and function evaluations, so the quadrature is efficient.

This quadrature method can then be used to generate the nonclassical orthogonal polynomials for weight functions containing logarithms and obtain true Gaussian quadratures for these weights. Two algorithms for the two types of logarithmic weights that incorporate these methods are given in the following paper.

Categories and Subject Descriptors: G.1.4 [**Numerical Analysis**]: Quadrature and Numerical Differentiation (F.2.1)—*Gaussian quadrature*

General Terms: Algorithms

Additional Key Words and Phrases: Gauss-Chebyshev quadrature, Gauss-Jacobi quadrature, Gauss-Laguerre quadrature, Gauss-Legendre quadrature, Gauss-type quadrature, logarithmic integrals, Maple V5.1 symbolic algebra system, Mehler quadrature, orthogonal polynomials

1. INTRODUCTION

The high-accuracy evaluation of integrals with logarithmic singularities at end points can be a difficult numerical problem. The recommended procedure [Zwillinger 1992, Chapter 28] of changing variables to remove the singularity is a useful analytic method. The numerical methods resulting from this procedure generally depend on detailed information about the integrand and have neither the accuracy nor flexibility of the methods that are derived here.

In this paper, we develop first a procedure to determine exactly (throughout this paper, this means up to machine accuracy) quadrature methods for evaluating

Authors' addresses: James S. Ball, University of Utah, Department of Physics, Salt Lake City, UT 84112-0830, USA, FAX: +1 801 581 6256, Tel: +1 801 581 8397. Nelson H. F. Beebe, University of Utah, Department of Mathematics, 155 S 1400 E RM 233, Salt Lake City, UT 84112-0090, USA, FAX: +1 801 585 1640, Tel: +1 801 581 5254.

integrals of polynomials of the following forms:[1]

$$\int_0^\infty dx\, x^\alpha e^{-x} \ln x\, f(x) \qquad (\alpha > -1) \tag{1}$$

and

$$\int_{-1}^1 dx\, (1-x)^\alpha (1+x)^\beta \ln(1+x) f(x) \qquad (\alpha > -1, \quad \beta > -1). \tag{2}$$

For polynomials of order $2N-1$ the quadrature method must have $2N$ parameters. For Gaussian quadrature, one makes use of the properties of orthogonal polynomials by choosing the N zeros of the N^{th} order polynomial as N parameters and the weights at each node as the other N parameters. If the recurrence relation of the polynomials is known these parameters can be easily calculated. The method proposed here is to use the fixed nodes of the related Gaussian quadrature and to express the integrals in terms of $2N$ parameters which can be calculated in a straightforward manner.

The final integrals are given in terms of the function and its derivative at the nodes. This is the result that one would obtain from a quadrature method using Hermitian interpolation for the function $f(x)$ (see, for example, Engels [1980, Chapter 7]). In this procedure, the weights are given in terms of integrals of polynomials of order $2N-1$ with the desired weight function. This leads to the "Catch 22"-like result that if one can accurately integrate polynomials of order $2N-1$, one can obtain a method for performing integrals of polynomials of order $2N-1$. Our calculation makes no use of Hermitian interpolation, but rather calculates the coefficients directly from the related Gaussian quadrature.

In the next section, the basic idea underlying these methods are developed with the necessary mathematical formulation. In Section 3, the quadrature method for the integral in Eq. (1) based on generalized Laguerre polynomials is presented. The mathematical details and the stability and accuracy of the resulting numerical treatment are discussed. This quadrature produces a set of orthogonal polynomials with a positive-definite weight function containing a linear logarithmic term. The recurrence relations for these polynomials is then used to produce a conventional Gaussian quadrature with this weight function. A Gaussian-like quadrature for Eq. (1) can then be obtained by combining the new Gaussian quadrature with the conventional generalized Gauss-Laguerre quadrature.

Finally, in Section 4, the quadrature method for Eq. (2) based on Jacobi polynomials is presented, following the same general procedure used in Section 3. In this case, a Gaussian quadrature is obtained for the following integral:

$$\int_{-1}^1 dx\, (1-x)^\alpha (1+x)^\beta \ln(\frac{1+x}{2}) f(x) = 2^{1+\alpha+\beta} \int_0^1 dy (1-y)^\alpha y^\beta \ln(y) f(2y-1). \tag{3}$$

Rescaling the coefficient in the log term from y to σy adds a constant term to the log, producing an extra nonlogarithmic integral that can be evaluated by conventional Gauss-Jacobi quadrature.

[1] In this paper, equations are numbered only if they are subsequently referred to in the text, or are cross-referenced from the companion paper or its computer programs.

The implementation, testing, and documentation of the associated software for the quadrature methods are provided in a companion article [Beebe and Ball 20xx].

2. DERIVATION OF THE QUADRATURE METHOD

We begin by considering the following integral:

$$\int_a^b dx\,(x+x_0)^\alpha g(x)f(x) \tag{4}$$

where

$$w(\alpha,x) = (x+x_0)^\alpha g(x)$$

is the weight function for a set of orthogonal polynomials with a known recurrence formula and the function $g(x)$ is independent of α. We require that $(a+x_0) > 0$ so that $w(\alpha,x)$ is single valued. For the case that $f(x)$ is a polynomial of order $2N-1$ or less, the N^{th} order quadrature is exact. Jacobi polynomials and generalized Laguerre polynomials provide examples for the application of this method and are used to produce Gaussian quadratures for several differing weight functions and integration ranges. The method derived here is based on the following observation:

$$\begin{aligned}\frac{\partial}{\partial\alpha}\int_a^b dx\,(x+x_0)^\alpha g(x)f(x) \;&\equiv\; \frac{\partial}{\partial\alpha}\int_a^b dx\,e^{\alpha\ln(x+x_0)}g(x)f(x)\\ &= \int_a^b dx\,(x+x_0)^\alpha\ln(x+x_0)g(x)f(x).\end{aligned}$$

Here we have assumed that the integral is uniformly convergent so that the derivative can be moved inside the integral.

The Gaussian quadrature based on these polynomials is as follows:

$$\int_a^b dx\,(x+x_0)^\alpha g(x)f(x) \approx \sum_{i=1}^{N} W_i(\alpha)f(x_i(\alpha)). \tag{5}$$

The integral is evaluated exactly, provided that $f(x)$ is a polynomial of order $2N-1$ or less. The fact that the nodes x_i and the weights W_i depend on α has been made explicit in Eq. (5). The Gaussian quadrature for the logarithmic integral is then obtained by taking the derivative of the right-hand side of Eq. (5) as follows:

$$\begin{aligned}&\int_a^b dx\,(x+x_0)^\alpha\ln(x+x_0)g(x)f(x) \approx\\ &\qquad\sum_{i=1}^{N}\Big[\frac{dW_i(\alpha)}{d\alpha}f(x_i(\alpha)) + W_i(\alpha)\frac{dx_i(\alpha)}{d\alpha}f'(x_i(\alpha))\Big].\end{aligned} \tag{6}$$

The quantities required for this quadrature are the weights and nodes and their derivatives with respect to α.

Let us begin by reviewing the procedure for obtaining the weights and nodes of the Gaussian quadrature. We choose the normalization of the orthogonal polynomials so that they are orthonormal. The orthogonality relation satisfied by these polynomials, denoted $P_n^\alpha(x)$, is

$$\int_a^b dx\,(x+x_0)^\alpha g(x) P_n^\alpha(x) P_m^\alpha(x) = \delta_{nm}.$$

with the usual Kronecker δ notation. The three-term recurrence formula satisfied by these polynomials is

$$x P_n^\alpha(x) = A_{n+1}^\alpha P_{n+1}^\alpha(x) + B_n^\alpha P_n^\alpha(x) + A_n^\alpha P_{n-1}^\alpha(x). \tag{7}$$

The Jacobi matrix for these polynomials is obtained by arranging the first N of these coefficients in an $N \times N$ matrix, $\mathbf{T}_N$, as follows:

$$\mathbf{T}_N(\alpha) = \begin{pmatrix} B_0^\alpha & A_1^\alpha & 0 & 0 & \cdots & 0 \\ A_1^\alpha & B_1^\alpha & A_2^\alpha & 0 & \cdots & 0 \\ 0 & A_2^\alpha & B_2^\alpha & A_3^\alpha & \cdots & 0 \\ . & . & . & . & \cdots & . \\ . & . & . & . & \cdots & . \\ . & . & . & . & \cdots & . \\ . & . & . & . & \cdots & A_{N-1}^\alpha \\ 0 & 0 & 0 & \cdots & A_{N-1}^\alpha & B_{N-1}^\alpha \end{pmatrix}. \tag{8}$$

The eigenvalues of $\mathbf{T}_N(\alpha)$ are the nodes, $x_i(\alpha)$, for the Gaussian quadrature and the transpose of the i^{th} column eigenvector is

$$\mathbf{V}_i^T(\alpha) = [P_0^\alpha(x_i(\alpha)), P_1^\alpha(x_i(\alpha)), \ldots, P_{N-1}^\alpha(x_i(\alpha))].$$

The key observation here is that the eigenvalues can be computed in $\mathcal{O}(N^2)$ operations using any standard tridiagonal-matrix eigenvalue solver, and that the eigenvector solution, which is normally an $\mathcal{O}(N^3)$ process, can be accomplished in $\mathcal{O}(N^2)$ operations using the three-term recurrence relations.

The weights can be calculated from the eigenvectors as follows [Wilf 1962]:

$$(W_i(\alpha))^{-1} = S_i^\alpha = \sum_{n=0}^{N-1} [P_n^\alpha(x_i(\alpha))]^2 = A_N^\alpha P_{N-1}^\alpha(x_i(\alpha)) P_N^{\alpha\prime}(x_i(\alpha)) \tag{9}$$

where

$$P_N^{\alpha\prime}(x_i(\alpha)) = \left.\frac{dP_n^\alpha(x)}{dx}\right|_{x=x_i(\alpha)}.$$

The evaluation of the sum given on the right of Eq. (9) is obtained by taking the appropriate limit of the Christoffel-Darboux identity [Luke 1977, p. 272, Eq. (23)] satisfied by these polynomials. In our numerical treatment, the expression in terms of the sum proved more robust and is therefore used in the following treatment.

To obtain the derivatives of the nodes with respect to α, introduce $\eta \ll \alpha$, and consider the Taylor expansion near α

$$\mathbf{T}_N(\alpha+\eta) = \mathbf{T}_N(\alpha) + \eta \mathbf{T}'_N(\alpha) + \mathcal{O}(\eta^2)$$

where

$$\mathbf{T}'_N(\alpha) = \frac{d\mathbf{T}_N(\alpha)}{d\alpha}.$$

First-order perturbation theory (see, for example, Mathews and Walker [1970, Chapter 10]) can now be used to determine the eigenvalues of $\mathbf{T}_N(\alpha+\eta)$:

$$x_i(\alpha+\eta) = x_i(\alpha) + \eta \mathbf{V}_i^T(\alpha)\mathbf{T}_N'(\alpha)\mathbf{V}_i(\alpha)/S_i^\alpha + \mathcal{O}(\eta^2)$$

with the result that

$$\begin{aligned}\frac{dx_i(\alpha)}{d\alpha} &= \mathbf{V}_i^T(\alpha)\mathbf{T}_N'(\alpha)\mathbf{V}_i(\alpha)/S_i^\alpha \\ &= W_i(\alpha)\sum_{n=0}^{N-1} P_n^\alpha(x_i(\alpha))[A_n^{\alpha\prime} P_{n-1}^\alpha(x_i(\alpha)) + B_n^{\alpha\prime} P_n^\alpha(x_i(\alpha)) + A_{n+1}^{\alpha\prime} P_{n+1}^\alpha(x_i(\alpha))].\end{aligned}$$

Here the primes indicate the derivative with respect to α, and as required by the recurrence formula, $P_{-1}^\alpha(x) = 0$.

To evaluate $dW_i(\alpha)/d\alpha$, the sum defined in Eq. (9) is used:

$$\frac{dW_i(\alpha)}{d\alpha} = -\frac{dS_i^\alpha}{d\alpha}/(S_i^\alpha)^2 = -2(W_i(\alpha))^2 \sum_{n=0}^{N-1} P_n^\alpha(x_i(\alpha))\frac{dP_n^\alpha(x_i(\alpha))}{d\alpha}.$$

While a recurrence formula for the total derivative of $P_n^\alpha(x_i(\alpha))$ can easily be derived from Eq. (7), this involves $dx_i(\alpha)/d\alpha$, the quantity that we seek to compute. A more efficient procedure is to express this quantity in terms of partial derivatives. This allows the simultaneous use of the recurrence formulas to evaluate the four sums necessary to obtain the weights for the logarithmic quadrature formula. If we define

$$\phi_n^\alpha(x) = \frac{\partial P_n^\alpha(x)}{\partial \alpha}$$

and

$$P_n^{\alpha\prime}(x) = \frac{dP_n^\alpha(x)}{dx},$$

then

$$\frac{dP_n^\alpha(x_i(\alpha))}{d\alpha} = \phi_n^\alpha(x_i(\alpha)) + P_n^{\alpha\prime}(x_i(\alpha))\frac{dx_i(\alpha)}{d\alpha}.$$

The recurrence formulas for these functions that are derived from Eq. (7) are

$$xP_n^{\alpha\prime}(x) = A_{n+1}^\alpha P_{n+1}^{\alpha\prime}(x) + B_n^\alpha P_n^{\alpha\prime}(x) + A_n^\alpha P_{n-1}^{\alpha\prime}(x) - P_n^\alpha(x) \tag{10}$$

and

$$\begin{aligned}x\phi_n^\alpha(x) &= A_{n+1}^{\alpha\prime} P_{n+1}^\alpha(x) + B_n^{\alpha\prime} P_n^\alpha(x) + A_n^{\alpha\prime} P_{n-1}^\alpha(x) + \\ &\quad A_{n+1}^\alpha \phi_{n+1}^\alpha(x) + B_n^\alpha \phi_n^\alpha(x) + A_n^\alpha \phi_{n-1}^\alpha(x).\end{aligned}$$

For many polynomials, a simple expression for $P_n^{\alpha\prime}(x)$ exists, and is in fact used for the two types of quadrature algorithms discussed in the next sections. The recurrence relations can be used simultaneously to calculate the four sums

$$S_i^\alpha = \sum_{n=0}^{N-1} [P_n^\alpha(x_i(\alpha))]^2, \tag{11}$$

$$\frac{dx_i(\alpha)}{d\alpha}S_i^\alpha = \sum_{n=0}^{N-1} P_n^\alpha(x_i(\alpha))[A_n^{\alpha\prime}P_{n-1}^\alpha(x_i(\alpha)) + B_n^{\alpha\prime}P_n^\alpha(x_i(\alpha)) + A_{n+1}^{\alpha\prime}P_{n+1}^\alpha(x_i(\alpha))], \quad (12)$$

$$Q_i^\alpha = \sum_{n=0}^{N-1} P_n^\alpha(x_i(\alpha))\phi_n^\alpha(x_i(\alpha)), \quad (13)$$

and

$$R_i^\alpha = \sum_{n=0}^{N-1} P_n^\alpha(x_i(\alpha))P_n^{\alpha\prime}(x_i(\alpha)) \quad (14)$$

necessary for the quadrature.

If we now define

$$\delta W_i(\alpha) = \frac{dW_i(\alpha)}{d\alpha} = -2(W_i(\alpha))^2(Q_i^\alpha + R_i^\alpha \frac{dx_i(\alpha)}{d\alpha}) \quad (15)$$

and

$$\delta x_i(\alpha) = W_i(\alpha)\frac{dx_i(\alpha)}{d\alpha}, \quad (16)$$

then the final expression for the quadrature of Eq. (6) is

$$\int_a^b dx\,(x+x_0)^\alpha \ln(x+x_0)g(x)f(x) \approx \sum_{i=1}^{N}[\delta W_i(\alpha)f(x_i(\alpha)) + \delta x_i(\alpha)f'(x_i(\alpha))]. \quad (17)$$

All of the calculations in the procedure above have $\mathcal{O}(N^2)$ operations.

3. LOG QUADRATURE BASED ON GENERALIZED GAUSS-LAGUERRE QUADRATURE

This section deals with obtaining a quadrature formula for integrals of the following form:

$$\int_0^\infty dx\, x^\alpha e^{-x} \ln x\, f(x) \qquad (\alpha > -1).$$

Clearly, in Eq. (4) and Eq. (6) we have $x_0 = 0$, $g(x) = e^{-x}$ and $w(\alpha, x) = x^\alpha e^{-x}$.

The related polynomials are the orthonormal versions of generalized Laguerre polynomials

$$\mathcal{L}_n^\alpha(x) = \sqrt{\frac{n!}{\Gamma(n+\alpha+1)}}L_n^\alpha(x), \quad (18)$$

where $L_n^\alpha(x)$ are the conventional Laguerre polynomials [Luke 1977, p. 273]. The recurrence coefficients and their derivatives with respect to α are as follows:

$$A_n^\alpha = -\sqrt{n(n+\alpha)}, \quad (19)$$

$$A_n^{\alpha\prime} = -\frac{1}{2}\sqrt{\frac{n}{(n+\alpha)}} = \frac{1}{2}A_n^\alpha/(n+\alpha), \quad (20)$$

$$B_n^\alpha = (2n + 1 + \alpha), \tag{21}$$

and

$$B_n^{\alpha\prime} = 1.$$

From Eq. (18) and the standard formula for the derivative of a generalized Laguerre polynomial [Abramowitz and Stegun 1964, p. 783, §22.8.6],

$$x \frac{d}{dx} L_n^{(\alpha)} = n L_n^{(\alpha)} - (\alpha + n) L_{n-1}^{(\alpha)},$$

the formula for evaluating $d\mathcal{L}_n^\alpha(x)/dx$ is

$$\frac{d\mathcal{L}_n^\alpha(x)}{dx} = \frac{1}{x}[n\mathcal{L}_n^\alpha(x) + A_n^\alpha \mathcal{L}_{n-1}^\alpha(x)].$$

Finally, the starting values for using the various recurrence formulas are provided by the following formula

$$\mathcal{L}_0^\alpha(x) = \frac{1}{\sqrt{\Gamma(1+\alpha)}} \qquad (\alpha > -1), \tag{22}$$

which is readily derived from Eq. (18) and the fact that $L_0^\alpha(x) = 1$ [Abramowitz and Stegun 1964, p. 785, §22.11.16].

Carrying out the calculations of the previous section then produces the $2N$ coefficients needed to evaluate integrals of the form given in Eq. (1). After the coefficients have been determined (an $\mathcal{O}(N^2)$ procedure), the actual computation of the integral requires N evaluations of the function, and N evaluations of its derivative, at the nodes, together with $2N - 1$ additions and $2N$ multiplications.

If the evaluation of the derivative is difficult or not possible, the procedure described below allows the evaluation of the integral in terms of $2N + 1$ evaluations of the function $f(x)$ with a quadrature that is exact if the function is a polynomial of order $2N - 2$ or less.

The procedure is the following: we first choose a weight function that is simply related to the logarithmic weight used above and that is positive definite so that it can be used to define a set of orthogonal polynomials. The log quadrature of the previous section can then be used to calculate the recurrence coefficients for this new set of polynomials. From this, the Gaussian quadrature defined by a new weight function and a new set of polynomials can be obtained.

While there are many possible choices for weight functions that contain $\ln x$, we use the inequality

$$x - 1 - \ln x \geq 0 \qquad (x > 0)$$

and choose what appears to be the simplest weight function that is positive definite:

$$w(\alpha, x) = (x - 1 - \ln x) x^\alpha e^{-x}. \tag{23}$$

From Eq. (5) and Eq. (6), the quadrature for this weight is

$$\int_0^\infty dx\, x^\alpha e^{-x} (x - 1 - \ln x) f(x) \approx$$
$$\sum_{i=1}^{N} \left\{ [W_i(\alpha)(x_i(\alpha) - 1) - \delta W_i(\alpha)] f(x_i(\alpha)) - \delta x_i(\alpha) f'(x_i(\alpha)) \right\}. \tag{24}$$

It proves numerically useful to work with monic (highest-order coefficient of unity) orthogonal polynomials, $M_n^\alpha(x)$, for which the corresponding three-term recurrence relation is

$$M_{n+1}^\alpha(x) = (x - b_n^\alpha)M_n^\alpha(x) - a_n^\alpha M_{n-1}^\alpha(x) \tag{25}$$

with initial conditions $M_{-1}^\alpha(x) = 0$ and $M_0^\alpha(x) = 1$. This iterative method for obtaining a set of orthogonal polynomials was apparently first suggested by Stieltjes [1884].

Multiplication of the recurrence by $M_{n-1}^\alpha(x)$ and by $M_n^\alpha(x)$, followed by integration with respect to the weight function, produces the recurrence coefficients as

$$a_n^\alpha = t_n^\alpha / t_{n-1}^\alpha \tag{26}$$

$$b_n^\alpha = s_n^\alpha / t_n^\alpha \tag{27}$$

where the right-hand sides require the zeroth and first moments:

$$t_n^\alpha = \int_0^\infty dx\, w(\alpha, x)(M_n^\alpha(x))^2 \tag{28}$$

$$s_n^\alpha = \int_0^\infty dx\, w(\alpha, x)(M_n^\alpha(x))^2 x \tag{29}$$

The integrands are positive, so both the moments and the recurrence coefficients are positive. The moments themselves can be computed accurately with our quadrature method involving functions and their derivatives.

The diagonal of the Jacobi matrix Eq. (8) is $b_0^\alpha, b_1^\alpha, \ldots$, and the off-diagonal is $\sqrt{a_1^\alpha}, \sqrt{a_2^\alpha}, \ldots$. As before, a_0^α is not needed in the Jacobi matrix, but we define it anyway from Eq. (26) by arbitrarily choosing the initial condition $t_{-1}^\alpha = 1$. Our software makes the $(a_n^\alpha, b_n^\alpha, s_n^\alpha, t_n^\alpha)$ values available to the user.

The orthonormal polynomials that satisfy the three-term recurrence relation, Eq. (7), can be recovered from the monic polynomials by

$$\Lambda_n^\alpha(x) = M_n^\alpha(x) / \sqrt{t_n^\alpha}$$

but we do not need them further here.

In the numeric quadrature used for evaluating the moments in Eq. (29), care must be taken to ensure that the quadrature order is large enough to provide 'exact' results. For the Laguerre case here, this is *two* more than the order of the base quadrature, and for the Jacobi case in the next section, *one* more.

The idea of evaluating the necessary integrals using a known set of moments for polynomials x^n fails because cancellation between individual terms destroys all accuracy at surprisingly small values of n. Indeed, Gautschi [1968, pp. 256–257] reported:

> [The condition number] κ_n grows at least at a rate essentially equal to … $\exp(3.5255 \cdots n)$. … Computing Christoffel numbers on the interval $(0, 1)$ from given moments is therefore about as ill-conditioned as the inversion of Hilbert matrices!

Better results have been obtained by Gautschi [1990] using a variety of methods including orthogonal reduction and the Stieltjes procedures. In these cases, the accuracy of the new quadrature obtained from the Jacobi matrix cannot be better than the accuracy of the quadrature method used to obtain the recurrence coefficients.

In our case, the application of the Stieltjes procedure to obtain the new Jacobi matrix introduces very little error for the following reason: the quadrature given in Eq. (24) allows us to evaluate the necessary integrals exactly, because the functions to be integrated are polynomials, and because both integrands are either positive definite or x times a positive definite function, there is little or no loss of accuracy. This method has the further advantage that all of the polynomials are evaluated at the same nodes. Therefore a step requires only using the recurrence formula to calculate the new polynomial and its derivative at each node.

Once the Jacobi matrix of the desired dimension is obtained, the weights and nodes for this Gaussian quadrature can be calculated by the methods given in Section 2. If we denote the new nodes $y_i(\alpha)$ and the new weights $Z_i(\alpha)$, we have obtained the following Gaussian quadrature:

$$\int_0^\infty dx\, x^\alpha e^{-x}(x-1-\ln x)f(x) \approx \sum_{i=1}^{N} Z_i(\alpha) f(y_i(\alpha)). \tag{30}$$

Utilizing this new Gaussian quadrature now means that integrals of the form of Eq. (1) can be obtained from Eq. (30) by subtracting the following integral:

$$\int_0^\infty dx\, x^\alpha e^{-x}(x-1)f(x) \tag{31}$$

This integral can easily be evaluated by generalized Gauss-Laguerre quadrature. This evaluation requires $2N+1$ function evaluations: $N+1$ at the nodes of the original generalized Gauss-Laguerre quadrature, and N at the nodes of the new Gaussian quadrature defined by Eq. (30).

In summary, the quadrature of Eq. (1) can be obtained in either of two ways:

$$\begin{aligned}\int_0^\infty dx\, x^\alpha e^{-x} \ln x\, f(x) \;&\approx\; \sum_{i=1}^{N}[\delta W_i(\alpha) f(x_i(\alpha)) + \delta x_i(\alpha) f'(x_i(\alpha))] &(32)\\ &\approx\; \sum_{i=1}^{N}[W_i(\alpha)(x_i(\alpha)-1) f(x_i(\alpha)) - Z_i(\alpha) f(y_i(\alpha))]. &(33)\end{aligned}$$

The nonlogarithmic case can be obtained from the weights and nodes of either method by:

$$\int_0^\infty dx\, x^\alpha e^{-x} f(x) \;\approx\; \sum_{i=1}^{N} W_i(\alpha) f(x_i(\alpha)). \tag{34}$$

4. LOG QUADRATURE BASED ON GAUSS-JACOBI QUADRATURE

This section deals with obtaining a Gaussian quadrature for integrals of the following form:

$$\int_{-1}^{1} dx\, (1-x)^\alpha (1+x)^\beta \ln(1+x) f(x) \qquad (\alpha > -1, \quad \beta > -1).$$

In this section, it is convenient to treat β as the parameter to be differentiated to produce the logarithmic factor, rather than the α of the previous sections. This interchange is necessary to produce the standard notation for the Jacobi polynomials. Clearly, in Eq. (4) we have $x_0 = 1$, $g(x) = (1-x)^\alpha$ and $w(\beta, x) = (1-x)^\alpha(1+x)^\beta$. The related polynomials are the orthonormal versions of Jacobi polynomials

$$\mathcal{P}_n^{(\alpha,\beta)}(x) = \sqrt{\frac{n!(2n+\alpha+\beta+1)\Gamma(n+\alpha+\beta+1)}{2^{\alpha+\beta+1}\Gamma(n+\alpha+1)\Gamma(n+\beta+1)}}P_n^{(\alpha,\beta)}(x), \tag{35}$$

where $P_n^{(\alpha,\beta)}(x)$ are the conventional Jacobi polynomials [Luke 1977, pp. 273–283]. The recurrence coefficients and their derivatives are as follows:

$$A_n^{(\alpha,\beta)} = \frac{2}{2n+\alpha+\beta}\sqrt{\frac{(n)(n+\alpha)(n+\beta)(n+\alpha+\beta)}{(2n+1+\alpha+\beta)(2n-1+\alpha+\beta)}}, \tag{36}$$

$$\begin{aligned} A_n^{(\alpha,\beta)\prime} = & \frac{-A_n^{(\alpha,\beta)}}{2}\left(\frac{(\alpha+\beta)}{(n+\alpha+\beta)(2n+\alpha+\beta)} + \right. \\ & \left. \frac{2n(\beta-\alpha)+\beta^2-\alpha^2+1}{[(2n+\alpha+\beta)^2-1](n+\beta)}\right) \end{aligned} \tag{37}$$

$$B_n^{(\alpha,\beta)} = -\frac{\alpha^2-\beta^2}{(2n+2+\alpha+\beta)(2n+\alpha+\beta)}, \tag{38}$$

and

$$B_n^{(\alpha,\beta)\prime} = -B_n^{(\alpha,\beta)}\left(\frac{2\beta}{\alpha^2-\beta^2} + \frac{1}{2n+\alpha+\beta+2} + \frac{1}{2n+\alpha+\beta}\right). \tag{39}$$

Here, $A_n^{(\alpha,\beta)\prime}$ and $B_n^{(\alpha,\beta)\prime}$ are derivatives with respect to β.

The formula for evaluating $d\mathcal{P}_n^{(\alpha,\beta)}(x)/dx$ is

$$\begin{aligned} & \frac{d\mathcal{P}_n^{(\alpha,\beta)}(x)}{dx} = \\ & \frac{1}{1-x^2}[(n\frac{(\alpha-\beta)}{2n+\alpha+\beta} - x)\mathcal{P}_n^{(\alpha,\beta)}(x) + A_n^{(\alpha,\beta)}(2n+\alpha+\beta+1)\mathcal{P}_{n-1}^{(\alpha,\beta)}(x)]. \end{aligned} \tag{40}$$

Finally, the starting values for using the various recurrence formulas are provided by the following formula:

$$\mathcal{P}_0^{(\alpha,\beta)}(x) = \sqrt{\frac{\Gamma(\alpha+\beta+2)}{\Gamma(1+\alpha)\Gamma(1+\beta)2^{\alpha+\beta+1}}}. \tag{41}$$

This follows from Eq. (35) and the fact that $P_0^{(\alpha,\beta)}(x) = 1$ [Abramowitz and Stegun 1964, p. 785, §22.11.1].

Carrying out the calculations of Section 2 then produces the $2N$ coefficients needed to evaluate integrals of the form given in Eq. (2). As in the previous section, performing the integral requires the evaluation of the function and its derivative at the nodes of the underlying Gauss-Jacobi quadrature.

If the evaluation of the derivative is difficult or not possible, we can follow the procedure of the previous section to produce a quadrature that allows the evaluation of the integral employing $2N$ evaluations of the function $f(x)$.

In this case, the simplest weight function that is positive definite is

$$\begin{aligned} w(\alpha,\beta,x) &= (1-x)^\alpha(1+x)^\beta(\ln(2)-\ln(1+x)) \\ &= -(1-x)^\alpha(1+x)^\beta\ln(\frac{1+x}{2}) \qquad (|x|<1). \end{aligned} \tag{42}$$

From the results obtained above, the quadrature for this weight is

$$\int_{-1}^{1} dx\,(1-x)^\alpha(1+x)^\beta\ln(\frac{1+x}{2})f(x) \approx \sum_{i=1}^{N}[(\delta W_i(\alpha,\beta)-\ln(2)W_i(\alpha,\beta))f(x_i(\alpha,\beta))+\delta x_i(\alpha,\beta)f'(x_i(\alpha,\beta))]. \tag{43}$$

Here we have replaced $\ln((1+x)/2)$ by $\ln(1+x)-\ln(2)$. The Gaussian-like quadrature derived above,

$$\int_{-1}^{1} dx\,(1-x)^\alpha(1+x)^\beta\ln(1+x)f(x) \approx \sum_{i=1}^{N}[\delta W_i(\alpha,\beta)f(x_i(\alpha,\beta))+\delta x_i(\alpha,\beta)f'(x_i(\alpha,\beta))], \tag{44}$$

is used to evaluate the integral containing $\ln(1+x)$ and the integral proportional to $\ln(2)$ is evaluated by generalized Gauss-Jacobi quadrature,

$$\int_{-1}^{1} dx\,(1-x)^\alpha(1+x)^\beta f(x) \approx \sum_{i=1}^{N} W_i(\alpha,\beta)f(x_i(\alpha,\beta)). \tag{45}$$

Integration of the latter form is sometimes known as *Mehler* quadrature. The special case $\alpha=\beta=0$ is *Gauss-Legendre* quadrature, and the case $\alpha=\beta=-1/2$ is *Gauss-Chebyshev* quadrature.

When no function derivatives are available, the Stieltjes' procedure of the previous section can be used to produce the nodes and weights of the Gaussian quadrature defined by the weight function in Eq. (42). If we denote the new nodes $y_i(\alpha,\beta)$ and the new weights $Z_i(\alpha,\beta)$, we have obtained a Gaussian quadrature formula for the integral in Eq. (3):

$$\int_{-1}^{1} dx\,(1-x)^\alpha(1+x)^\beta\ln(\frac{1+x}{2})f(x) \;\approx\; -\sum_{i=1}^{N} Z_i(\alpha,\beta)f(y_i(\alpha,\beta)). \tag{46}$$

A change in the coefficient of $(1+x)$ within the logarithm adds a constant to the log term which can be integrated using conventional Gauss-Jacobi quadrature. Utilizing this new Gaussian quadrature now means that integrals of the form of Eq. (2) (with an arbitrary coefficient in the log) can be obtained by $2N$ function evaluations: N at the nodes of the ordinary Gauss-Jacobi quadrature, Eq. (45), and N at the nodes of the new Gaussian quadrature defined by Eq. (46).

In summary, the quadrature of Eq. (2) can be obtained either by Eq. (44), or by

$$\int_{-1}^{1} dx\,(1-x)^{\alpha}(1+x)^{\beta}\ln(1+x)f(x) \approx \sum_{i=1}^{N}[\ln(2)W_i(\alpha,\beta)f(x_i(\alpha,\beta)) - Z_i(\alpha,\beta)f(y_i(\alpha,\beta))]. \tag{47}$$

Similarly, the quadrature for Eq. (3) can be computed either by Eq. (46), or by Eq. (43). The nonlogarithmic case can be obtained from the weights and nodes of either method by Eq. (45).

5. SIMPLE TESTS OF LOG QUADRATURES

To test our methods and compare the results obtained by the generalized Gauss-Laguerre quadrature and Gauss-Jacobi quadrature, we consider the following integrals with known closed-form values:

$$\int_{0}^{\infty} dx\,x^{\alpha}e^{-x}\ln(x)\,x^{n} \;=\; \Gamma(\alpha+n+1)\psi(\alpha+n+1) \tag{48}$$

$$\int_{-1}^{1} dx\,(1-x)^{\alpha}(1+x)^{\beta}\ln(\frac{1+x}{2})\,(1-x)^{n} = 2^{\alpha+\beta+n+1}\frac{\Gamma(\beta+1)\Gamma(\alpha+n+1)}{\Gamma(\alpha+\beta+n+2)}(\psi(\beta+1)-\psi(\alpha+\beta+n+2)) \tag{49}$$

These are integrals of n^{th} order polynomials with the logarithmic weight functions, which should be evaluated exactly by our basic method and by the Gaussian quadrature we have obtained from the basic methods. This process has three sources of error:

(1) The first and most important are the parameters needed for the quadrature. The values for the nodes are more accurate than the various weights because the latter are obtained by summing N terms. Somewhat more accurate values for the weights of the classical quadrature can be obtained by matrix methods, such as Gautschi [1994] used in his Gauss-Jacobi quadrature algorithm.

(2) The second source of error is the calculation of function and derivative values, x_i^n and $(1-x_i)^n$. This error is smaller, but not negligible, and grows with n.

(3) The final source of error in the evaluation of the special functions $\Gamma(x)$ and $\psi(x)$. These quantities enter into the calculation of the recurrence coefficients, and also into the analytic expressions for the integrals.

All of these errors can be reduced by using higher precision, and it is advisable to do so when possible. Our software implementation described in [Beebe and Ball 20xx] includes both double- and quadruple-precision versions, even though not all systems support the latter. It is then straightforward to provide an interface with the double-precision routine names that calls the quadruple-precision routines internally.

In Table I, we give the results for the integral in Eq. (48) for 20 nodes and $\alpha = -15/16$ (exactly representable in binary floating-point arithmetic) to emphasize the case of small x for small n. The derivative form, Eq. (32), adds about one decimal digit of accuracy. As one would expect, the generalized Gauss-Laguerre quadrature results, Eq. (34), that include the log term in the function are poor when the contribution from small x is important, but generally improve as n increases.

Table I. Magnitudes of relative error in double-precision quadrature of log-Laguerre test Eq. (48) for $\alpha = -15/16$ with 20 nodes. The worst-case errors (in boldface) correspond to about 11 and 45 units in the last place in columns 2 and 3 of each table, respectively. A similar test with a 128-bit quadruple-precision version of our programs had worst-case errors of 21 and 90 units in the last place. For comparison, the NAG Library double-precision adaptive-quadrature routine, `d01amf()`, required about 1900 function evaluations for $n = 0$ to reach a relative error of about 2.50e−07; for $n \geq 1$, it produced relative errors below 5 units in the last place with 700 to 1000 function evaluations. The larger error at $n = 39$ for Eq. (33) is expected, since that method is exact only to polynomial order $2N - 2$.

n	Eq. (32)	Eq. (33)	Eq. (34)	n	Eq. (32)	Eq. (33)	Eq. (34)
0	8.91e−16	0.00e+00	6.86e−01	20	2.94e−16	6.02e−15	2.94e−16
1	2.16e−15	1.80e−15	1.38e−01	21	5.76e−16	5.76e−15	6.91e−16
2	0.00e+00	1.87e−15	2.09e−03	22	8.62e−16	6.90e−15	1.03e−15
3	2.21e−16	4.42e−16	2.95e−05	23	1.23e−15	7.27e−15	1.23e−15
4	2.15e−16	8.59e−16	9.92e−07	24	1.35e−15	8.09e−15	1.69e−15
5	3.54e−16	2.13e−15	5.15e−08	25	1.77e−15	8.85e−15	1.99e−15
6	8.67e−16	2.73e−15	3.60e−09	26	1.95e−15	**9.91e−15**	1.95e−15
7	1.49e−15	3.13e−15	3.18e−10	27	2.37e−15	8.47e−15	2.37e−15
8	1.41e−15	3.62e−15	3.43e−11	28	2.38e−15	7.13e−15	2.57e−15
9	1.62e−15	2.94e−15	4.39e−12	29	**2.46e−15**	4.58e−15	2.46e−15
10	1.48e−15	2.47e−15	6.57e−13	30	2.31e−15	2.80e−15	2.19e−15
11	1.13e−15	1.88e−15	1.12e−13	31	2.31e−15	3.85e−16	2.31e−15
12	6.54e−16	1.83e−15	2.28e−14	32	2.10e−15	2.75e−15	2.10e−15
13	3.36e−16	2.35e−15	4.53e−15	33	1.68e−15	6.48e−15	1.68e−15
14	2.00e−16	3.39e−15	1.40e−15	34	1.12e−15	7.71e−15	1.12e−15
15	2.21e−16	3.98e−15	0.00e+00	35	8.11e−16	9.15e−15	8.11e−16
16	0.00e+00	4.58e−15	0.00e+00	36	6.29e−16	9.64e−15	8.39e−16
17	2.23e−16	5.13e−15	2.23e−16	37	0.00e+00	9.60e−15	1.11e−15
18	0.00e+00	5.54e−15	0.00e+00	38	1.58e−16	7.91e−15	1.01e−14
19	0.00e+00	5.88e−15	0.00e+00	39	6.60e−16	1.48e−10	**1.95e−13**

We test our log-Jacobi methods by evaluating the integral in Eq. (49), which has polynomials of order n times the basic weight function. In this case, we have chosen powers of $(1 - x)$ so that the strength of the log singularity is not suppressed by large values of n. The results are given in Table II. Here, as in Table I, we have chosen α and β values to emphasize the logarithmic singularity.

As expected, the results for both of our methods were very good, with the derivative form, Eq. (44), affording about one more decimal digit of accuracy, whereas the results for the conventional Gauss-Jacobi quadrature, Eq. (45), are completely unacceptable.

Table II. Magnitudes of relative error in double-precision quadrature of log-Jacobi test Eq. (49) for $\alpha = \beta = -15/16$ with 20 nodes. The worst-case errors (in boldface) correspond to about 32 and 91 units in the last place in columns 2 and 3 of each table respectively. A similar test with a 128-bit quadruple-precision version of our programs had worst-case errors of 83 and 21 units in the last place. For comparison, the NAG Library double-precision adaptive-quadrature routine, `d01amf()`, produced relative errors below 7 units in the last place, at the expense of 180 to 560 function evaluations.

n	Eq. (44)	Eq. (47)	Eq. (45)
0	**7.09e−15**	1.74e−14	5.66e−01
1	6.72e−15	1.83e−14	5.69e−01
2	6.74e−15	1.86e−14	5.71e−01
3	6.56e−15	1.86e−14	5.72e−01
4	6.36e−15	1.85e−14	5.73e−01
5	6.37e−15	1.87e−14	5.74e−01
6	6.18e−15	1.85e−14	5.75e−01
7	6.19e−15	1.88e−14	5.76e−01
8	5.99e−15	1.86e−14	5.77e−01
9	5.79e−15	1.86e−14	5.77e−01
10	5.59e−15	1.86e−14	5.78e−01
11	5.59e−15	1.88e−14	5.78e−01
12	5.18e−15	1.87e−14	5.79e−01
13	5.19e−15	1.87e−14	5.79e−01
14	4.98e−15	1.87e−14	5.80e−01
15	4.99e−15	1.89e−14	5.80e−01
16	4.78e−15	1.85e−14	5.81e−01
17	4.58e−15	1.87e−14	5.81e−01
18	4.37e−15	1.85e−14	5.82e−01
19	4.58e−15	1.90e−14	5.82e−01

n	Eq. (44)	Eq. (47)	Eq. (45)
20	4.38e−15	1.90e−14	5.82e−01
21	4.17e−15	1.90e−14	5.83e−01
22	4.17e−15	1.92e−14	5.83e−01
23	4.18e−15	1.90e−14	5.83e−01
24	3.97e−15	1.90e−14	5.84e−01
25	3.76e−15	1.90e−14	5.84e−01
26	3.76e−15	1.92e−14	5.84e−01
27	3.97e−15	1.95e−14	5.84e−01
28	3.98e−15	1.97e−14	5.85e−01
29	3.35e−15	1.91e−14	5.85e−01
30	3.35e−15	1.93e−14	5.85e−01
31	3.35e−15	1.95e−14	5.85e−01
32	3.14e−15	1.95e−14	5.86e−01
33	3.15e−15	1.95e−14	5.86e−01
34	3.15e−15	1.99e−14	5.86e−01
35	3.36e−15	1.99e−14	5.86e−01
36	3.15e−15	**2.02e−14**	5.87e−01
37	2.94e−15	1.99e−14	5.87e−01
38	2.52e−15	1.97e−14	5.87e−01
39	2.73e−15	2.00e−14	**5.87e−01**

6. RELATED WORK

Krylov and Pal'tsev [1971] developed quadrature formulas, and produced extensive tables of quadrature nodes and weights, for these four integrals:

$$\int_0^1 dy\, y^\alpha \ln(e/y) f(y)$$

$$\int_0^1 dy\, y^\beta \ln(e/y) \ln(e/(1-y)) f(y)$$

$$\int_0^1 dy\; \ln(1/y) f(y)$$

$$\int_0^\infty dx\, x^\beta e^{-x} \ln(1+1/x) f(x)$$

The first and third of these are special cases of our Eq. (3). The remaining ones do not have a simple relation to ours.

Danloy [1973] derived quadrature formulas for an integral similar to Eq. (3). He gave an $\mathcal{O}(N)$ formula, and a somewhat better-conditioned $\mathcal{O}(N^2)$ formula, for the determination of *each* of the coefficients in a three-term recurrence relation, from which the total work to determine the integration weights and nodes is $\mathcal{O}(N^2)$ and $\mathcal{O}(N^3)$, respectively. Although his paper finishes with a note that applications to other quadratures are under investigation, we have not been able to find any further published work by him in this area.

Since the algorithms proposed here require $\mathcal{O}(N^2)$ total work for the determination of the weights and nodes, we have not pursued Danloy's approach for our case with limits 0 to ∞.

In a landmark article, Gautschi [1994] combined work published in dozens of papers into a comprehensive package, *ORTHPOL*, for the generation of orthogonal polynomials and Gauss-type quadrature rules. An example of numerical problems with logarithmic weights is provided by absorbing the logarithms in our Eq. (1) and Eq. (2) into the function $f(x)$. His Fortran code could be applied to the quadratures discussed in this paper. However, the integrand would then contain a logarithmic singularity at one end point, and numerical accuracy would suffer.

Gautschi [1994, p. 32] in his Example 3.2 gives a single example of a logarithmic weight function of the form $t^\sigma \ln(1/t)$ for the interval $(0, 1]$ with the constraint $\sigma > -1$. This is a special case of our Eq. (3), with $\alpha = 0$ and $\beta = \sigma$. He tabulates 25-digit values of the recurrence coefficients for $\sigma = -0.5, 0, 0.5$. Using a special quadruple-precision version (about 34 decimal digits) of our program, we were able to obtain agreement for all but the last digit in the recurrence coefficients given in his Table III, except for β_{99}, where two final digits differ. This appears to be consistent with Gautschi's estimates of the error in these coefficients given in his Table IX.

7. CONCLUSION

A general quadrature method for integration of functions with weights that are logarithms multiplied by Jacobi or Laguerre polynomial weights has been developed. These methods are then used to generate new orthogonal polynomials with weights closely related to these weight-functions. Two new Gaussian quadratures are derived from these polynomials. Finally, the general integrals considered can be evaluated as a sum of two Gaussian quadratures: the new one derived here, and the related classical Gaussian quadrature.

Our Jacobi quadrature procedure can be generalized by interchanging α and β. When these are added together, one obtains a quadrature for the Jacobi weight multiplied by $\ln[(1-x)^\sigma(1+x)^\rho]$ for arbitrary values of σ and ρ, both > -1.

In principle, the method described here could be extended to higher powers of logarithms, using higher-order perturbation theory. Danloy [1973, p. 865] also considered such powers, but his method led to a factor of $\mathcal{O}(N)$ more work for each additional power.

Also, our method could be generalized by using second-order perturbation theory to produce a method for log-squared weights that would require knowledge of the function and its first and second derivatives at the nodes. Each new power of the logarithm requires one higher order of perturbation theory and one more derivative of the function. The fact that higher-order perturbation theory quickly becomes very complicated makes this process of doubtful utility. Perhaps a better method could be developed by applying the basic idea to the polynomials with logarithmic weights resulting in a quadrature for log-squared, which could then be used to calculate a new set of polynomials with log-squared weight. While repeating this process is straightforward, the fact that the Jacobi matrix is only known numerically means that the matrix $T'(\alpha)$ cannot be calculated analytically. The necessary numerical differentiation may be possible, at least for a few higher orders, but is certainly a source of error.

REFERENCES

ABRAMOWITZ, M. AND STEGUN, I. A., Eds. 1964. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables.* Applied mathematics series, vol. 55. U.S. Department of Commerce, Washington, DC, USA. Tenth printing, with corrections (December 1972).

BEEBE, N. H. F. AND BALL, J. S. 20xx. Algorithm *xxx*: Quadruple-precision $\gamma(x)$ and $\psi(x)$ functions for real arguments. *ACM Transactions on Mathematical Software ??,* ?? (????), ??–??

DANLOY, B. 1973. Numerical construction of Gaussian quadrature formulas for $\int_0^1 (-\operatorname{Log} x) \cdot x^\alpha \cdot f(x) \cdot dx$ and $\int_0^\infty E_m(x) \cdot f(x) \cdot dx$. *Mathematics of Computation 27,* 124 (Oct.), 861–869.

ENGELS, H. 1980. *Numerical Quadrature and Cubature.* Academic Press, New York, NY, USA.

GAUTSCHI, W. 1968. Construction of Gauss-Christoffel quadrature formulas. *Mathematics of Computation 22,* 102 (Apr.), 251–270.

GAUTSCHI, W. 1990. Orthogonal polynomials: Theory and practice. In *Orthogonal polynomials: theory and practice: proceedings of the NATO Advanced Study Institute on "Orthogonal Polynomials and Their Applications," the Ohio State University, Columbus, Ohio, U.S.A., May 22, 1989–June 3, 1989*, P. Nevai and M. Ismail, Eds. NATO ASI series. Series C, Mathematical and physical sciences, vol. 294. Kluwer Academic Publishers Group, Norwell, MA, USA, and Dordrecht, The Netherlands, 181–216.

GAUTSCHI, W. 1994. Algorithm 726: ORTHPOL — A package of routines for generating orthogonal polynomials and Gauss-type quadrature rules. *ACM Transactions on Mathematical Software 20,* 1 (Mar.), 21–62. See remark [Gautschi 1998].

GAUTSCHI, W. 1998. Remark on Algorithm 726: ORTHPOL — A package of routines for generating orthogonal polynomials and Gauss-type quadrature rules. *ACM Transactions on Mathematical Software 24,* 3 (Sept.), 355–355. See [Gautschi 1994].

KRYLOV, V. I. AND PAL'TSEV, A. A. 1971. *Tables for Numerical Integration of Functions with Logarithmic and Power Singularities.* Israel Program for Scientific Translation, Jerusalem, Israel. Translation of Russian original: *Tablitsy dlia chislennogo integrirovaniia funktsii s logarifmicheskimi i stepennymi osobennostiami*, Izdatel'stvo "Nauka i Teknika", Minsk 1967.

LUKE, Y. L. 1977. *Algorithms for the Computation of Mathematical Functions.* Academic Press, New York, NY, USA.

MATHEWS, J. AND WALKER, R. L. 1970. *Mathematical methods of physics*, Second ed. W. A. Benjamin, Inc., New York, NY, USA.

STIELTJES, T. J. 1884. Quelques recherches sur la théorie des quadratures dites mécaniques. *Ann. Sci. Ec. Norm. Paris, Sér. 3 1*, 409–426.

WILF, H. S. 1962. *Mathematics for the Physical Sciences.* Wiley, New York, NY, USA.

ZWILLINGER, D. 1992. *Handbook of Integration.* Jones and Bartlett Publishers, Inc., One Exeter Plaza, Boston, MA 02116, USA.

Received July 2000; revised June 2004; accepted XXXX YYYY