MATHEMATICS OF COMPUTATION, VOLUME 27, NUMBER 124, OCTOBER 1973

Numerical Construction of Gaussian Quadrature Formulas for

$$\int_0^1 (-\mathrm{Log}\, x)\cdot x^\alpha \cdot f(x)\cdot dx \quad \text{and} \quad \int_0^\infty E_m(x)\cdot f(x)\cdot dx$$

By Bernard Danloy

Abstract. Most nonclassical Gaussian quadrature rules are difficult to construct because of the loss of significant digits during the generation of the associated orthogonal polynomials. But, in some particular cases, it is possible to develop stable algorithms. This is true for at least two well-known integrals, namely

$$\textstyle\int_0^1 -(\mathrm{Log}\, x)\cdot x^\alpha\cdot f(x)\cdot dx \quad \text{and} \quad \int_0^\infty E_m(x)\cdot f(x)\cdot dx.$$

A new approach is presented, which makes use of known classical Gaussian quadratures and is remarkably well-conditioned since the generation of the orthogonal polynomials requires only the computation of discrete sums of positive quantities. Finally, some numerical results are given.

1. Introduction. Let $w(x)$ be a nonnegative weight function on (a, b) such that all its moments

$$\mu_k = \int_a^b w(x)\cdot x^k\cdot dx, \qquad k = 0, 1, 2, \cdots, \tag{1.1}$$

exist. The n-point Gaussian quadrature rule associated with $w(x)$ and (a, b) is that uniquely defined linear functional

$$G_n\cdot f \equiv \sum_{i=1}^n \lambda_i\cdot f(x_i) \tag{1.2}$$

which satisfies

$$G_n\cdot f = \int_a^b w(x)\cdot f(x)\cdot dx \tag{1.3}$$

whenever f is a polynomial of degree $\leqq 2n - 1$.

It is a well-known result [8] that the Gaussian abscissas x_i are the roots of the polynomials orthogonal on (a, b) with respect to $w(x)$, and that the associated coefficients λ_i, called Christoffel constants, can also be expressed in terms of these polynomials. A direct exploitation of these results is still the most widely recommended procedure, even though alternative approaches have been suggested by Rutishauser [7], Golub and Welsch [6]. Actually, all methods make direct or indirect use of the orthogonal polynomials and thus require their generation if they are not known.

Gautschi [3] was the first to consider the numerical stability of the whole problem; in fact, he fully elucidated the ill-conditioning of the basic problem, namely that of

Received July 18, 1972.

AMS (MOS) subject classifications (1970). Primary 65D30, 65G05; Secondary 33A65.

Key words and phrases. Gaussian quadrature rules, numerical condition, orthogonal polynomials.

solving the algebraic system

$$\sum_{i=1}^{n} \lambda_i \cdot x_i^k = \mu_k , \qquad 0 \leqq k \leqq 2n - 1. \tag{1.4}$$

More precisely, the loss of significant digits occurs during the orthogonalization of the sequence 1, x, x^2, $\cdots$, i.e., while generating the orthogonal polynomials. According to this result, Gautschi distinguished the following two cases:

(a) The classical case. The orthogonal polynomials are known, and the problem is well-conditioned.

(b) The nonclassical case. The orthogonal polynomials are not known and have to be generated.

The latter case is obviously the more frequent, and practical algorithms would be very welcome. Actually, Gautschi suggested two procedures which are numerically stable:

(a) The first method is based on an approximate discretization of the orthogonality relation [3]. This is equivalent to computing by some approximate quadrature rule the coefficients of the recurrence relation satisfied by the orthogonal polynomials. Convergence has been proved under reasonable assumptions but may be rather slow.

(b) The second method makes use of "modified moments" [5]. Instead of the sequence of monomials 1, x, x^2, $\cdots$, one can orthogonalize any set of linearly independent polynomials. In several cases, a suitable choice of this set strongly improves the numerical condition of the problem. The method does not involve any approximation, but requires much skill for its practical implementation.

The next sections consider two specific cases of Gaussian quadrature, namely

$$\int_0^1 (-\text{Log } x) \cdot x^\alpha \cdot f(x) \cdot dx \quad \text{and} \quad \int_0^\infty E_m(x) \cdot f(x) \cdot dx;$$

the central result is an exact discretization of the orthogonality relation, which enables us to generate the orthogonal polynomials in a very stable way.

2. Orthogonal Polynomials.

Given $w(x)$ on (a, b), we can define the sequence $\{p_k(x)\}_{k=0}^{\infty}$ of orthogonal polynomials with leading coefficients equal to one. These polynomials satisfy the recurrence relation [9]

$$p_{k+1} = (x - \alpha_k)p_k(x) - \beta_k \cdot p_{k-1}(x), \qquad k \geqq 1, \tag{2.1}$$

with

$$p_0(x) = 1; \qquad p_1(x) = x - \alpha_0. \tag{2.2}$$

If we put

$$h_k = \int_a^b w(x) \cdot \{p_k(x)\}^2 \cdot dx, \qquad k = 0, 1, 2, \cdots , \tag{2.3}$$

the coefficients of the recurrence relation are given by

$$\beta_k = h_k/h_{k-1}, \qquad k = 1, 2, 3, \cdots , \tag{2.4}$$

$$\alpha_k = \frac{1}{h_k} \int_a^b w(x) \cdot x \cdot \{p_k(x)\}^2 \cdot dx, \qquad k = 0, 1, 2, \cdots . \tag{2.5}$$

If a is finite, Eq. (2.1) can be replaced by the system

$$\begin{aligned} p_{k+1}(x) &= (x - a)\cdot\pi_k(x) - (\gamma_k/h_k)\cdot p_k(x), \\ \pi_{k+1}(x) &= p_{k+1}(x) - (h_{k+1}/\gamma_k)\cdot\pi_k(x), \end{aligned} \qquad k = 0, 1, 2, \cdots, \tag{2.6}$$

with

$$\gamma_k = \int_a^b w(x)\cdot(x - a)\cdot\{\pi_k(x)\}^2\cdot dx, \tag{2.7}$$

where $\{\pi_k(x)\}_{k=0}^{\infty}$ is the set of polynomials with leading coefficients one and orthogonal on (a, b) with respect to the weight function $w(x)\cdot(x - a)$.

A similar result holds if b is finite.

Since (2.1) and (2.6) are equivalent, it follows immediately that

$$\alpha_k = \gamma_k/h_k + h_k/\gamma_{k-1} + a. \tag{2.8}$$

The problem of generating the orthogonal polynomials $p_k(x)$ is to determine the coefficients of the recurrence relation, i.e., to compute numerically the quadratures (2.3) and (2.5) (or (2.7)).

It is important to realize that the use of moments should be entirely bypassed because of ill-conditioning. A lower estimate of the condition number for the classical procedure has been given by Gautschi [3]; in most cases, one must expect it to grow as fast as $(33.97)^n/n^2$.

It is thus vital to find a better-conditioned approach to work out a numerical method for the exact computation of integrals of the type:

$$\int_a^b w(x)\cdot\rho(x)\cdot dx, \tag{2.9}$$

where $\rho(x)$ is a polynomial. A partial solution will now be given.

3. Logarithmic Weight Functions. As stated above, the construction of Gaussian rules for

$$\int_0^1 (-\operatorname{Log} x)\cdot x^\alpha\cdot f(x)\cdot dx \qquad (\alpha > -1)$$

requires a computationally suitable expression for

$$J(\rho) = \int_0^1 (-\operatorname{Log} x)\cdot x^\alpha\cdot\rho(x)\cdot dx \qquad (\alpha > -1), \tag{3.1}$$

where $\rho(x)$ is a polynomial.

Elementary transformations yield

$$\begin{aligned} J(\rho) &= \int_0^1 dx\cdot x^\alpha\cdot\rho(x)\int_x^1 \frac{dt}{t} \\ &= \int_0^1 \frac{dt}{t}\int_0^t dx\cdot x^\alpha\cdot\rho(x) \\ &= \int_0^1 dt\cdot t^\alpha\int_0^1 du\cdot u^\alpha\cdot\rho(ut) \end{aligned} \tag{3.2}$$

and thus, if $\{A_i, u_i\}_{i=1}^N$ is the N-point Gaussian quadrature formula for $\int_0^1 x^\alpha \cdot f(x) \cdot dx$, it follows that

$$J(\rho) = \sum_{i=1}^N \sum_{j=1}^N A_i \cdot A_j \cdot \rho(u_i u_j) \tag{3.3}$$

for any polynomial $\rho(x)$ of degree $\leqq 2N - 1$.

Thus, for any $k < N$, we have

$$h_k = \sum_{i=1}^N \sum_{j=1}^N A_i \cdot A_j \cdot \{p_k(u_i u_j)\}^2 \tag{3.4}$$

and

$$\alpha_k = \frac{1}{h_k} \sum_{i=1}^N \sum_{j=1}^N A_i A_j u_i u_j \{p_k(u_i u_j)\}^2. \tag{3.5}$$

A recursive computation of the above quantities is thus trivial since, with h_0, α_0, h_1, α_1, $\cdots$, h_{k-1}, α_{k-1} previously obtained, the evaluation of h_k and α_k only requires the values $p_k(u_i u_j)$ which are easily computed from (2.1). Similarly, (3.3) yields

$$\gamma_k = \sum_{i=1}^N \sum_{j=1}^N A_i A_j u_i u_j \{\pi_k(u_i u_j)\}^2 \tag{3.6}$$

and the whole sequence h_0, γ_0, h_1, γ_1, h_2, γ_2, $\cdots$ can be generated by a recursive use of (2.6), (3.4) and (3.6). It should be stressed that this process is remarkably stable with respect to rounding errors since all the terms of (3.4), (3.5) and (3.6) are positive.

Another useful expression for h_k and γ_k can be obtained as follows: Integration by parts yields

$$\int_0^1 (-\text{Log } x) \cdot \{x^{\alpha+1} \cdot \rho(x)\}' \cdot dx = \int_0^1 x^\alpha \cdot \rho(x) \cdot dx. \tag{3.7}$$

Thus, for all $k < N$,

$$\begin{aligned} \sum_{i=1}^N A_i \cdot \{p_k(u_i)\}^2 &= \int_0^1 (-\text{Log } x) \cdot x^\alpha \cdot \{(\alpha + 1)p_k^2(x) + 2x \cdot p_k'(x) \cdot p_k(x)\} \cdot dx \\ &= (2k + \alpha + 1) \cdot h_k . \end{aligned} \tag{3.8}$$

Similarly,

$$\sum_{i=1}^N A_i \cdot u_i \cdot \{\pi_k(u_i)\}^2 = (2k + \alpha + 2) \cdot \gamma_k . \tag{3.9}$$

This second approach may seem more attractive than the first one originating from (3.3), since the amount of work is proportional to N instead of N^2. However, its expected stability is a little weaker: h_k does actually depend on h_0, h_1, $\cdots$, h_{k-1} and γ_0, γ_1, $\cdots$, γ_{k-1}, and varies with the rounding errors which affect them. Nevertheless, to terms of first order, the representation (2.3) of h_k is independent of such perturbations; this is also true if (3.4) is used, but is no longer valid in the case of (3.8). Although it was not observed for moderate values of k, (3.8) and (3.9) might thus suffer from a slight instability.

TABLE 1

$w(x) = \text{Log}(1/x), 0 < x < 1$

Coefficients of the recurrence relation

n	α_n	β_n
0	.250000000000	.0000000000000
1	.464285714286	.0486111111111
2	.485482446456	.0586848072562
3	.492103081871	.0607285839189
4	.495028498758	.0614820201969
5	.496579511644	.0618408095319
6	.497501301305	.0620390629544
7	.498094018204	.0621599191583
8	.498497801978	.0622389376716
9	.498785322656	.0622933886799
10	.498997353167	.0623324775066
11	.499158221678	.0623614734838
12	.499283180216	.0623835683595
13	.499382187671	.0624007864342
14	.499461972097	.0624144615353
15	.499527212427	.0624255013029
16	.499581244730	.0624345406235
17	.499626499806	.0624420343142
18	.499664782928	.0624483150597
19	.499697457641	.0624536307243

4. Generalization. The above results can easily be extended to more general cases involving either a more general weight function or a smaller range of integration:

(a) $$\int_0^1 (-\text{Log}\, x)^m \cdot x^\alpha \cdot f(x) \cdot dx, \qquad m = 1, 2, 3, \cdots .$$

The transformations performed under (3.2) can be repeated m times and yield a $(m + 1)$-tuple quadrature. This approach is rather expensive and probably impractical if m is large. If $m = 2$, one gets

(4.1) $$\begin{aligned} \int_0^1 (-\text{Log}\, x)^2 \cdot x^\alpha \cdot \rho(x) \cdot dx &= 2 \int_0^1 dt \cdot t^\alpha \int_0^1 du \cdot u^\alpha \int_0^1 dx \cdot x^\alpha \cdot \rho(xut) \\ &= 2 \sum_{i=1}^N \sum_{j=1}^N \sum_{k=1}^N A_i A_j A_k \rho(u_i u_j u_k). \end{aligned}$$

Obviously, (4.1) enjoys the same interesting properties as (3.3) and leads to a very well-conditioned procedure:

(b) $$\int_0^E (-\text{Log}\, x) \cdot x^\alpha \cdot f(x) \cdot dx, \qquad 0 < E \leqq 1.$$

Here too, a very similar treatment produces

(4.2) $$\begin{aligned} &\int_0^E (-\text{Log}\, x) \cdot x^\alpha \cdot \rho(x) \cdot dx \\ &= E^{\alpha+1} \cdot \left\{ (-\text{Log}\, E) \int_0^1 x^\alpha \cdot \rho(Ex)\, dx + \int_0^1 dx \cdot x^\alpha \int_0^1 dt \cdot t^\alpha \rho(Ext) \right\} \end{aligned}$$

and application of a suitable Gaussian formula again yields a discrete sum of positive terms.

TABLE 2

$w(x) = \text{Log}(1/x),\ 0 < x < 1$

10-point Gaussian quadrature

x_i	λ_i
.904263096219(-2)	.120955131955
.539712662225(-1)	.186363542564
.135311824639	.195660873278
.247052416287	.173577142183
.380212539609	.135695672995
.523792317972	.936467585381(-1)
.665775205517	.557877273514(-1)
.794190416012	.271598108992(-1)
.898161091219	.951518260284(-2)
.968847988719	.163815763360(-2)

20-point Gaussian quadrature

x_i	λ_i
.258832795592(-2)	.431427521332(-1)
.152096623496(-1)	.753837099086(-1)
.385365503721(-1)	.930532674517(-1)
.721816138158(-1)	.101456711850
.115460526488	.103201762056
.167442856275	.100022549805
.226983787260	.932597993003(-1)
.292754960941	.840289528720(-1)
.363277429858	.732855891300(-1)
.436957140091	.618503369137(-1)
.512122594679	.504166044385(-1)
.587064044915	.395513700052(-1)
.660073413315	.296940778958(-1)
.729484083930	.211563153554(-1)
.793709671987	.141237329390(-1)
.851280892789	.866097450433(-2)
.900879680854	.471994014620(-2)
.941369749129	.215139740396(-2)
.971822741075	.719728214653(-3)
.991538081439	.120427676330(-3)

5. The Exponential Integral as Weight Function. In the theory of radiations [1], one encounters quadratures of the form

$$\int_0^\infty E_m(x)\cdot f(x)\cdot dx \qquad (m > 0), \tag{5.1}$$

where $E_m(x)$ is the exponential integral

$$E_m(x) = \int_1^\infty \frac{e^{-xt}}{t^m}\,dt = x^{m-1}\int_x^\infty \frac{e^{-u}}{u^m}\,du. \tag{5.2}$$

All moments exist and the generation of the orthogonal polynomials only requires a convenient representation for

$$J(\rho) = \int_0^\infty E_m(x)\cdot \rho(x)\cdot dx. \tag{5.3}$$

TABLE 3

$$w(x) = E_1(x) = \int_1^\infty \frac{e^{-xt}}{t}\,dt, \qquad 0 < x < \infty$$

Coefficients of the recurrence relation

n	α_n	β_n
0	.500000000000	.000000000000
1	.230000000000(+1)	.416666666667
2	.423469387755(+1)	.261333333333(+1)
3	.619917288995(+1)	.675494793836(+1)
4	.817602876377(+1)	.128700101592(+2)
5	.101594224559(+2)	.209689946964(+2)
6	.121467630339(+2)	.310570066309(+2)
7	.141367011214(+2)	.431369541345(+2)
8	.161284556528(+2)	.572106684370(+2)
9	.181215390823(+2)	.732793862452(+2)
10	.201156292899(+2)	.913439870981(+2)
11	.221105037868(+2)	.111405121907(+3)
12	.241060033128(+2)	.133463287826(+3)
13	.261020104696(+2)	.157518874426(+3)
14	.280984365631(+2)	.183572193521(+3)
15	.300952131707(+2)	.211623499185(+3)
16	.320922865502(+2)	.241673001620(+3)
17	.340896138248(+2)	.273720877032(+3)
18	.360871603136(+2)	.307767274831(+3)
19	.380848976220(+2)	.343812322983(+3)

Elementary transformations yield

$$\begin{aligned} J(\rho) &= \int_0^\infty dx\cdot\rho(x)\cdot x^{m-1}\int_x^\infty dt\cdot e^{-t}/t^m \\ &= \int_0^\infty dt\cdot\frac{e^{-t}}{t^m}\int_0^t dx\cdot x^{m-1}\cdot\rho(x) \qquad (5.4)\\ &= \int_0^\infty dt\cdot e^{-t}\int_0^1 du\cdot u^{m-1}\cdot\rho(ut). \end{aligned}$$

Thus, if $\{A_i, u_i\}_{i=1}^N$ and $\{B_i, v_i\}_{i=1}^N$ are N-point Gaussian formulas for, respectively, $\int_0^1 x^{m-1}\cdot f(x)\cdot dx$ and $\int_0^\infty e^{-x}\cdot f(x)\cdot dx$, it follows that

$$J(\rho) = \sum_{i=1}^N \sum_{j=1}^N A_i B_j \rho(u_i v_j) \qquad (5.5)$$

for any polynomial $\rho(x)$ of degree $\leqq 2N-1$.

Using (5.5), it is then easy to compute h_k, γ_k, α_k, and β_k ($k \leqq N-1$), and well-conditioning is guaranteed since the summation involves only positive quantities.

As was done in the third section, another useful expression can be obtained: Integration by parts gives

$$\int_0^\infty E_m(x)\cdot\{x\cdot\rho(x)\}'\cdot dx = -\int_0^\infty x\cdot E_m'(x)\cdot\rho(x)\cdot dx. \qquad (5.6)$$

But from (5.1) one can derive

TABLE 4

$$w(x) = E_1(x) = \int_1^\infty \frac{e^{-xt}}{t}\,dt, \qquad 0 < x < \infty$$

10-point Gaussian quadrature

x_i	λ_i
.762404872624(-1)	.485707599602
.525005762690	.357347318586
.143921959617(+1)	.127267083774
.286324173848(+1)	.263265788641(-1)
.484664684765(+1)	.314097760897(-2)
.745940513320(+1)	.203866802942(-3)
.108101228402(+2)	.648957008433(-5)
.150823083551(+2)	.848669062576(-7)
.206311914914(+2)	.324960593979(-9)
.283693946254(+2)	.156050745676(-12)

20-point Gaussian quadrature

x_i	λ_i
.415731018684(-1)	.330068388136
.274239640181	.335018800621
.735213024633	.202727089842
.143646482057(+1)	.906794137819(-1)
.238684236390(+1)	.311926475044(-1)
.359494938617(+1)	.830681682460(-2)
.507042045480(+1)	.170519552143(-2)
.682474522008(+1)	.267192385286(-3)
.887199456612(+1)	.315225257509(-4)
.112296313102(+2)	.275116436420(-5)
.139195560950(+2)	.173736449646(-6)
.169695733188(+2)	.771970107104(-8)
.204155650908(+2)	.232856519358(-9)
.243048836079(+2)	.454955059531(-11)
.287019535563(+2)	.540355193304(-13)
.336981998188(+2)	.356730027303(-15)
.394313665944(+2)	.114479498566(-17)
.461284475183(+2)	.143415823640(-20)
.542229680449(+2)	.463374056292(-24)
.648259442473(+2)	.136239857196(-28)

(5.7) $$x \cdot E_m'(x) = (m - 1)\cdot E_m(x) - e^{-x}.$$

Combining (5.6) and (5.7) yields

(5.8) $$\int_0^\infty E_m(x)\cdot\{m\cdot\rho(x) + x\cdot\rho'(x)\}\cdot dx = \int_0^\infty e^{-x}\cdot\rho(x)\cdot dx.$$

Then, Gauss-Laguerre integration gives, for all $k < N$,

(5.9) $$\sum_{i=1}^{N} B_i\cdot\{p_k(v_i)\}^2 = \int_0^\infty E_m(x)\cdot\{m\cdot p_k^2(x) + 2x\cdot p_k'(x)\cdot p_k(x)\}\cdot dx$$
$$= (2k + m)h_k$$

and similarly

$$\sum_{i=1}^{N} B_i v_i \{\pi_k(v_i)\}^2 = (2k + m + 1)\gamma_k . \qquad (5.10)$$

The latter formulas are clearly attractive, but, for the same reason as the one explained at the end of the third section, they might be slightly less reliable than (5.5) if k is large.

6. Numerical Results. Using the above procedures, orthogonal polynomials of degree up to 40 were generated for both kinds of quadrature considered. The roots of these polynomials were then found by a standard procedure and the associated Christoffel constants were computed using a computationally optimized representation [2]. Listed below are the first twenty coefficients of the recurrence relation and the 10- and 20-point Gaussian formulas for $w(x) = \text{Log}(1/x)$, $0 < x < 1$, and for $w(x) = E_1(x)$, $0 < x < \infty$. Computations were performed with 48-bit floating-point arithmetic, but the last two decimal digits have been dropped since round-off might affect them.

7. Conclusion. The above results completely agree with those of Stroud and Secrest [8]. Some errors were found in the values published by Gautschi [4], but they affect only the last one or two digits and are likely to result either from the underlying approximation or from some numerical round-off.

It should be emphasized that we completely avoided unnecessary loss of significant digits and that, unlike Stroud and Secrest, we were able to get highly accurate results without any use of multiple-precision arithmetic. Our approach is at the same time rather inexpensive and completely reliable; it unfortunately depends on the specific quadrature considered, especially on the weight function, but seems to be the best procedure, whenever it is feasible. Its application to some other quadratures is presently under investigation.

Faculté Polytechnique, B.P.255
Université Nationale du Zaïre
Kinshasa XI, République du Zaïre, Africa

1. S. CHANDRASEKHAR, *The Transfer of Radiant Energy*, Clarendon Press, Oxford, 1953. Chap. II. MR **13**, 136.

2. B. DANLOY, *Improving Accuracy in the Computation of Christoffel Constants*, Université de Montréal, Département d'Informatique, Publication #80 (février 1972).

3. W. GAUTSCHI, "Construction of Gauss-Christoffel quadrature formulas," *Math. Comp.* v. 22, 1968, pp. 251–270. MR **37** #3755.

4. W. GAUTSCHI, "Algorithm 331, Gaussian quadrature formulas," *Comm. ACM*, v. 11, 1968, pp. 432–436.

5. W. GAUTSCHI, "On the construction of Gaussian quadrature rules from modified moments," *Math Comp.*, v. 24, 1970, pp. 242–260. MR **44** #2341a.

6. G. H. GOLUB & J. H. WELSCH, "Calculation of Gauss quadrature rules," *Math Comp.*, v. 23, 1969, pp. 221–230; addendum, ibid. v. 23, 1969, no. 106, loose microfiche suppl. A1–A10. MR **39** #6513.

7. H. RUTISHAUSER, "On a modification of the Q.D. algorithm with Graeffe-type convergence," *Proc. IFIP Congress Munich*, 1962, pp. 93–96.

8. A. H. STROUD & DON SECREST, *Gaussian Quadrature Formulas*, Prentice-Hall, Englewood Cliffs, N.J., 1966. MR **34** #2185.

9. G. SZEGÖ, *Orthogonal Polynomials*, 2nd rev. ed., Amer. Math. Soc. Colloq. Publ., vol. 23, Amer. Math Soc., Providence, R.I., 1959, Chap. 3. MR **21** #5029.