BIT 23 (1983), 209-216

HOW AND HOW NOT TO CHECK GAUSSIAN QUADRATURE FORMULAE*

WALTER GAUTSCHI

Department of Computer Sciences, Purdue University, West Lafayette, Indiana 47907, U.S.A.

Abstract.

We discuss characteristic difficulties inherent in the validation of Gaussian quadrature formulae, in particular the futility of moment-related tests. We propose instead more effective tests based on the recursion coefficients of the appropriate orthogonal polynomials and on the sum of the quadrature nodes.

1. Introduction.

The preparation of this note was prompted by the appearance, in the chemistry literature, of a 16-digit table of a Gaussian quadrature formula for integration with measure $d\lambda(t) = \exp(-t^3/3)dt$ on $(0, \infty)$, a table, which we suspected is accurate to only 1–2 decimal digits. How does one go about convincing a chemist, or anybody else for that matter, that his Gaussian quadrature formula is seriously defective?

The question is not as easy to answer as one might think at first, and is not without intrinsic interest, considering that the most obvious test – using the n-point quadrature rule to reproduce the first $2n$ moments of $d\lambda$ – is totally ineffective. The latter, of course, is a manifestation of the extreme ill-conditioning (if n is large) of the map from the first $2n$ moments to the n-point Gaussian quadrature rule (cf. [3, § 3.2]).

In Section 2 we present the nodes $\tau_\nu^{(n)}$ and weights $\lambda_\nu^{(n)}$ for the Gaussian formula

$$\int_0^\infty f(t)\exp(-t^3/3)dt = \sum_{\nu=1}^{n} \lambda_\nu^{(n)} f(\tau_\nu^{(n)}) + R_n(f) \tag{1.1}$$

with $n = 15$, on the one hand as published in the literature, and on the other as recomputed by us. In Section 3 we discuss two tests, both ineffective, designed to determine which of the two formulae is the more trustworthy one. More conclusive tests are described in Section 4 which not only allow us to decide in favor of one of the two formulae, but also to indicate the accuracy of each.

---

*) Work sponsored in part by the National Science Foundation under grant MCS-7927158A1.
Received September 13, 1982.

2. Two competing implementations of (1.1).

In Table 2.1 are listed the nodes and weights of the quadrature rule (1.1) as published in [5]. (Integers in parentheses denote decimal exponents.) They were obtained by applying a "product-difference algorithm" (a variant of Rutishauser's quotient-difference algorithm) of R. G. Gordon [4] to produce the J-fraction belonging to $d\lambda(t) = \exp(-t^3/3)dt$, starting from the moments $\mu_k = \int_0^\infty t^k d\lambda(t) = 3^{(k-2)/3} \times \Gamma((k+1)/3)$, $k = 0, 1, 2, \ldots$. The nodes $\tau_\nu^{(n)}$ can then be obtained as eigenvalues of a symmetric tridiagonal matrix (the Jacobi matrix

Table 2.1. *The Gauss formula* (1.1) *according to* [5].

ν	$\tau_\nu^{(15)}$	$\lambda_\nu^{(15)}$
1	1.457697817613696(−2)	3.805398607861561(−2)
2	8.102669876765460(−2)	9.622028412880550(−2)
3	2.081434595902250(−1)	1.572176160500219(−1)
4	3.944841255669402(−1)	2.091895332583340(−1)
5	6.315647839882239(−1)	2.377990401332924(−1)
6	9.076033998613676(−1)	2.271382574940649(−1)
7	1.210676808760832(0)	1.732845807252921(−1)
8	1.530983977242980(0)	9.869554247686019(−2)
9	1.861844587312434(0)	3.893631493517167(−2)
10	2.199712165681546(0)	9.812496327697071(−3)
11	2.543839804028289(0)	1.439191418328875(−3)
12	2.896173043105410(0)	1.088910025516801(−4)
13	3.262066731177372(0)	3.546866719463253(−6)
14	3.653371887506584(0)	3.590718819809800(−8)
15	4.102376773975577(0)	5.112611678291437(−11)

for $d\lambda$), and the weights $\lambda_\nu^{(n)}$ in terms of the first components of the associated eigenvectors. The procedure, thus, is a particular realization of the (ill-conditioned) map from the moments to the Gaussian quadrature rule. (The sensitivity to rounding errors of this procedure has been explicitly noted by Gordon, who suggests the use of double precision arithmetic or, better yet, exact integer arithmetic. It is not clearly stated by the author of [5] what computer, and what type of arithmetic, he has used.) Table 2.2 displays the same quadrature rule (1.1), produced, however, by an application of the "discretized Stieltjes procedure", in combination with a suitable partition of the interval $(0, \infty)$ into eight subintervals (cf. [3, Example 4.6]), to generate the required orthogonal polynomials. Essentially the same method as in Gordon [4] was then used to obtain the $\tau_\nu^{(n)}$ and $\lambda_\nu^{(n)}$. (The computation was carried out in double precision on the CDC 6500, using a relative error tolerance of 0.5×10^{-20} in the discretized Stieltjes procedure.) It is seen that the two tables agree only to about 1–2 decimal digits. Which one is correct?

Table 2.2. *The Gauss formula* (1.1) *recomputed.*

ν	$\tau_\nu^{(15)}$	$\lambda_\nu^{(15)}$
1	1.929765389638693(−2)	4.940830823126689(−2)
2	1.006599142226749(−1)	1.126586278069619(−1)
3	2.428468366694404(−1)	1.696700745266622(−1)
4	4.387642946878456(−1)	2.136246330297717(−1)
5	6.787965036904373(−1)	2.329324905722498(−1)
6	9.522620471509191(−1)	2.150021042138036(−1)
7	1.249165311141012(0)	1.596591146577856(−1)
8	1.561526358196975(0)	8.939650846589768(−2)
9	1.883496691223344(0)	3.512652914092340(−2)
10	2.213595570164661(0)	8.956321788320709(−3)
11	2.550023378308307(0)	1.353123731389520(−3)
12	2.895208615030500(0)	1.076566880888657(−4)
13	3.254368222416162(0)	3.781200408411502(−6)
14	3.639045691197643(0)	4.272835535767259(−8)
15	4.080805415015807(0)	7.218347932277564(−11)

3. Moment-related tests.

As already mentioned, the ability of the quadrature formula to reproduce the moments accurately is an unreliable test. This will now be documented by performing two tests, first a simple moment-reproducing test, then a more involved test based on Markov's remainder formula.

Test #1. Verify $R_n(f) = 0$ for $f(t) = t^k$, $\quad k = 0, 1, 2, \ldots, 2n-1.$

In other words, use the formulae in Tables 2.1 and 2.2 to check the identities (with $n = 15$)

$$\mu_k = \sum_{\nu=1}^{n} \lambda_\nu^{(n)} [\tau_\nu^{(n)}]^k, \qquad k = 0, 1, 2, \ldots, 2n-1, \tag{3.1}$$

where μ_k are the moments $\mu_k = \int_0^\infty t^k \exp(-t^3/3)dt = 3^{(k-2)/3} \times \Gamma((k+1)/3)$, $k = 0, 1, 2, \ldots$.

Results of Test #1. Using double precision on the CDC 6500 (which corresponds to a precision of about 29 significant decimal digits) to carry out the computations indicated in (3.1), we obtain for the relative errors $\varepsilon_k = |(\mu_k - \sum_{\nu=1}^{n} \lambda_\nu^* [\tau_\nu^*]^k)/\mu_k|$ the results shown in Table 3.1. In the second and third columns are listed the errors ε_k resulting from the quadrature formula $(\tau_\nu^*, \lambda_\nu^*)$ of

Table 3.1. *Relative errors* ε_k *observed in reproducing the moments* μ_k *by the quadrature formulae of Tables* 2.1 *and* 2.2.

k	Table 2.1	Table 2.2
0	4.56(−16)	2.02(−17)
1	8.92(−16)	4.66(−17)
2	3.20(−15)	1.51(−16)
3	6.49(−15)	2.45(−16)
4	1.02(−14)	3.21(−16)
.	.	.
.	.	.
.	.	.
29	7.48(−14)	4.46(−16)

Table 2.1 and Table 2.2, respectively. The maximum error in each case is attained for $k = 29$.

Discussion. It is rather remarkable that both quadrature rules of Section 2, even though they differ already in the first or second decimal digit, manage to compute the moments to about 15 correct decimal digits. The reason for this is an extreme case of *correlation of errors.* If we represent the correct quadrature rule by a point $g_0 \in \mathbb{R}^{2n}$, and the vector of moments by $m_0 \in \mathbb{R}^{2n}$, then $g_0 = M_n m_0$ for some (nonlinear) map $M_n: \mathbb{R}^{2n} \to \mathbb{R}^{2n}$ defined in a neighborhood of m_0. For sufficiently small perturbations Δm of m_0, one has $\Delta g \approx J^0_{M_n} \Delta m$, where $J^0_{M_n}$ is the Jacobian matrix of M_n at m_0. Therefore, a small sphere $S(m_0; \varepsilon)$ of moment vectors, with center at m_0 and radius ε, is mapped under M_n, approximately, into an ellipsoid $E(g_0; \varepsilon)$ centered at g_0, whose half axes have lengths $\varepsilon \cdot \sigma_i(J^0_{M_n})$, with $\sigma_i(J^0_{M_n})$ the singular values of $J^0_{M_n}$, and directions given by the (orthonormal) eigenvectors of $J^0_{M_n}(J^0_{M_n})^T$. In our case, the ellipsoid $E(g_0; \varepsilon)$ happens to be extremely elongated and flat, the largest singular value being approx. 5.78×10^{11}, and the smallest 1.88×10^{-18}! It is reasonable to assume that the computed Gauss formula g^* is the exact Gauss formula belonging to some moment vector $m^* \in S(m_0; \varepsilon)$, where ε is of the order of magnitude of the machine precision, or a few orders larger (to account for rounding errors in the computational process). Conversely, then, to g^* there corresponds $m^* \in S(m_0; \varepsilon)$, hence $\|m^* - m_0\| \leqq \varepsilon$, explaining the relatively high accuracy of m^*. *The correlation of errors thus consists in the fact that the computed formula* g^* *lies, approximately, in the extremely elongated and flat ellipsoid* $E(g^0; \varepsilon)$. This correlation can be confirmed by subjecting either of the two quadrature rules of Section 2 to random errors of given magnitude ϱ, and letting ϱ vary through 10^{-1}, 10^{-2}, 10^{-3}, ... One will find that the resulting relative errors in the moments are no longer of the order of magnitude 10^{-15}, but rather of the order of magnitude ϱ, or significantly larger. The correlation of errors has been broken!

The discussion just given has merit only in a qualitative, not quantitative,

sense, the reason being that the moments μ_k vary in a wide range (from about 1.29 for $k = 0$ to about 7.14×10^9 for $k = 29$) and that one really ought to analyze the propagation of errors under M_n consistently in terms of *relative* errors. This can be done by computing a carefully defined condition number (cond M_n)(m_0) for the map M_n at m_0 (see [3, § 3.2, especially Eq. (3.11)]), with the result that (cond M_n)(m_0) $= 1.26 \times 10^{17}$ (for $n = 15$). This means that relative errors in the moments μ_k of magnitude ε must be expected to translate into relative errors in the Gauss formula having magnitude $\approx 10^{17} \cdot \varepsilon$. It would appear, therefore, that the formula in Table 2.1, given that it is accurate to only 1–2 decimal digits (cf. Section 4), was computed in a precision of about 20 decimal digits, assuming that the moments were computed correctly to machine precision.

Test # 2. Use the given quadrature nodes $\tau_\nu^{(n)}$ and weights $\lambda_\nu^{(n)}$ (for $n = 15$) to first generate the coefficients α_k, β_k, $k = 0, 1, 2, \ldots, n-1$, in the recursion formula

$$\pi_{k+1}(t) = (t-\alpha_k)\pi_k(t) - \beta_k \pi_{k-1}(t), \quad k = 0, 1, 2, \ldots, \quad \pi_{-1}(t) = 0, \quad \pi_0(t) = 1, \tag{3.2}$$

for the associated (monic) orthogonal polynomials $\{\pi_k(\cdot\,; d\lambda)\}$, where $\beta_0 = \int_0^\infty d\lambda(t)$. This is easily done (see, e.g., [3, Eq. (3.7)]) and yields the Jacobi matrix

$$J_n = J_n(d\lambda) = \begin{bmatrix} \alpha_0 & \sqrt{\beta_1} & & 0 \\ \sqrt{\beta_1} & \alpha_1 & \sqrt{\beta_2} & \\ & \ddots & \ddots & \ddots \\ & & \ddots & \sqrt{\beta_{n-1}} \\ 0 & & \sqrt{\beta_{n-1}} & \alpha_{n-1} \end{bmatrix}. \tag{3.3}$$

Then, using the successive segments $J_1, J_2, \ldots, J_{n-1}$ of $J_n(d\lambda)$, generate (by methods already mentioned in Section 2) the nodes $\tau_\nu^{(k)}$ and weights $\lambda_\nu^{(k)}$ of the k-point Gaussian quadrature rule, $k = 1, 2, \ldots, n-1$, and check the identities

$$\mu_{2k} = \sum_{\nu=1}^{k} \lambda_\nu^{(k)} [\tau_\nu^{(k)}]^{2k} + \beta_0 \beta_1 \ldots \beta_k, \qquad k = 1, 2, \ldots, n-1. \tag{3.4}$$

(These follow readily from Markov's formula for the remainder term in Gaussian quadrature.)

Results of Test # 2. Using double precision as before, this test, too, disappointingly, is as unrevealing as Test # 1. Both quadrature rules of Section 2 confirm all identities in (3.4) to within a relative error of at most 7.24×10^{-14} and 4.36×10^{-16}, respectively.

4. Coefficient-based tests.

More effective are tests based on coefficients, either the coefficients in the three-term recurrence relation satisfied by the orthogonal polynomials, or coefficients of the orthogonal polynomials themselves. We propose two such tests.

Test # 3. Generate the coefficients α_k, β_k, $k = 0, 1, \ldots, n-1$, as in Test # 2; then check the relations

$$\left.\begin{aligned} \alpha_k &= D'_{k+1}/D_{k+1} - D'_k/D_k \\ \beta_k &= D_{k+1}D_{k-1}/D_k^2 \end{aligned}\right\} \quad k = 0(1)n-1 \text{ where} \tag{4.1}$$

$$D_0 = D_{-1} = 1,\ D_1 = \mu_0;\ D'_0 = 0,\ D'_1 = \mu_1; \tag{4.2}$$

$$D_k = \begin{vmatrix} \mu_0 & \mu_1 & \cdots & \mu_{k-1} \\ \mu_1 & \mu_2 & \cdots & \mu_k \\ \cdots & \cdots & \cdots & \cdots \\ \mu_{k-1} & \mu_k & \cdots & \mu_{2k-2} \end{vmatrix}, \quad D'_k = \begin{vmatrix} \mu_0 & \mu_1 & \cdots & \mu_{k-2} & \mu_k \\ \mu_1 & \mu_2 & \cdots & \mu_{k-1} & \mu_{k+1} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \mu_{k-1} & \mu_k & \cdots & \mu_{2k-3} & \mu_{2k-1} \end{vmatrix}, \quad k = 2, 3, \ldots.$$

Table 4.1. *Relative errors in the recursion coefficients α_k, β_k generated by the quadrature rules of Table 2.1 and Table 2.2.*

k	Table 2.1		Table 2.2	
	$\delta\alpha_k$	$\delta\beta_k$	$\delta\alpha_k$	$\delta\beta_k$
0	1.35(−15)	4.56(−16)	6.68(−17)	2.02(−17)
1	5.38(−15)	5.75(−15)	3.75(−17)	2.52(−16)
2	2.49(−15)	6.42(−15)	2.89(−17)	4.66(−19)
3	1.32(−13)	6.59(−14)	1.54(−16)	7.72(−17)
4	1.43(−12)	8.73(−13)	3.20(−16)	2.01(−16)
5	2.26(−11)	9.64(−12)	2.10(−16)	7.77(−16)
6	6.73(−10)	2.43(−10)	1.85(−16)	1.16(−15)
7	1.29(−8)	6.31(−9)	6.24(−16)	5.13(−16)
8	9.98(−8)	8.26(−8)	6.08(−16)	4.32(−16)
9	4.59(−7)	1.94(−7)	4.39(−16)	5.97(−16)
10	6.36(−6)	5.71(−6)	5.17(−16)	6.93(−16)
11	2.08(−4)	3.12(−5)	6.55(−16)	2.15(−15)
12	2.80(−3)	2.01(−3)	4.29(−14)	1.08(−14)
13	3.66(−2)	1.33(−3)	3.17(−13)	3.14(−13)
14	8.28(−2)	3.65(−1)	5.71(−12)	4.75(−13)

Results of Test # 3. Using the LINPACK double precision routine DGECO (see, e.g., [2, Ch. I]) to factor the Hankel matrices in (4.2), and computing the determinants $D_k, D'_k, k = 2, 3, \ldots$, as signed products of the diagonal elements of the upper triangular factors, one obtains the results shown in Table 4.1. Here,

$\delta\alpha_k$ and $\delta\beta_k$ are defined by $\delta\alpha_k = |(\alpha_k^* - \alpha_k)/\alpha_k|$, $\delta\beta_k = |(\beta_k^* - \beta_k)/\beta_k|$, where α_k^*, β_k^* are the recursion coefficients generated as in Test # 2 from the respective quadrature rule $(\tau_\nu^*, \lambda_\nu^*)$ in Table 2.1 and Table 2.2, while α_k, β_k are the recursion coefficients as computed from (4.1), (4.2).

Discussion. The map from the Gaussian quadrature formula $(\tau_\nu^{(n)}, \lambda_\nu^{(n)})_{\nu=1}^n$ to the recursion coefficients α_k, β_k, $k = 0, 1, \ldots, n-1$, is usually quite well-conditioned; see the discussion in [3, § 3.1]. In the present case, the appropriate condition number indeed computes to 28.0. This means that Test # 3 ought to be able not only to discriminate between a good and a bad quadrature formula, but also to determine, approximately, the accuracy of each. From the results in Table 4.1 it indeed becomes plausible that the quadrature rule of Table 2.1 is accurate to at most 1–2 decimal digits, while the one of Table 2.2 is accurate to at least 11 decimal digits.

Actually, our quadrature rule in Table 2.2 is probably accurate to all 16 digits shown. The reason why this is not evident from Table 4.1 is the fact that the Hankel matrices H_k and H_k' used in (4.2) also become ill-conditioned rather quickly, as k increases. Their triangular factorizations, therefore, suffer in accuracy accordingly. Fortunately, for the case at hand, double precision on the CDC 6500 is still sufficient to weather this progressive ill-conditioning. From the condition numbers furnished by the routine DGECO, and shown in Table 4.2, we can

Table 4.2. *Condition numbers of the Hankel matrices* H_k, H_k' *in* (4.2).

k	condH_k	condH_k'	k	condH_k	condH_k'
1	1.00(0)	1.00(0)	9	1.40(11)	2.90(11)
2	1.15(1)	7.18(0)	10	5.85(12)	1.29(13)
3	1.79(2)	1.48(2)	11	2.58(14)	6.02(14)
4	3.95(3)	4.17(3)	12	1.20(16)	2.93(16)
5	1.02(5)	1.33(5)	13	5.81(17)	1.48(18)
6	3.02(6)	4.61(6)	14	2.93(19)	7.76(19)
7	9.95(7)	1.72(8)	15	1.53(21)	4.20(21)
8	3.59(9)	6.85(9)			

see indeed that, in the worst case $k = 15$, at least 8–10 decimal digits are salvaged. However, it would be unreasonable to expect agreement in the recursion coefficients α_{14}, β_{14} to much more than 10 decimal digits. As it turned out (see Table 4.1), we observed agreement to about 11–12 decimal digits.

To make Test # 3 effective as a general test, one would normally have to consider the use of multiple-precision arithmetic. With the availability of precompilers, such as Augment [1], to convert single- or double-precision Fortran routines to multiple-precision routines, this would be quite feasible. However, we have not done so here.

For easy reference, we list in Table 4.3 what we believe are the correct values (to 16 decimals) of the coefficients α_k, β_k.

Table 4.3. *The recursion coefficients* α_k, β_k.

k	α_k	β_k
0	0.7290111329472270	1.2878993168540691
1	1.0422198256747441	0.2450009794174209
2	1.2537306422019648	0.3530735172799071
3	1.4061820889340039	0.4538065447547201
4	1.5304717088698266	0.5467091516329361
5	1.6371146876931010	0.6327914312656564
6	1.7313265280009314	0.7135915502415592
7	1.8162157284093990	0.7901716008181790
8	1.8938033162945061	0.8632766955003995
9	1.9654868263312374	0.9334529739837076
10	2.0322783394582394	1.0011143019264016
11	2.0949374105669606	1.0665830763064052
12	2.1540505128026898	1.1301163055170060
13	2.2100811161203424	1.1919228879829775
14	2.2634026387069418	1.2521754391488299

A simpler test, but one that checks only the nodes, is

Test #4. Compute the sum of the nodes, $s_n = \sum_{\nu=1}^{n} \tau_\nu^{(n)}$, and check against

$$s_n = D_n'/D_n, \tag{4.3}$$

where D_n, D_n' are the determinants defined in (4.2).

Results of Test #4. Using again the LINPACK routine DGECO, one finds

$$\begin{aligned} D_n'/D_n &= 25.7603125030, \\ s_n &= 25.4984452247 \quad \text{from Table 2.1,} \\ s_n &= 25.7603125030 \quad \text{from Table 2.2,} \end{aligned}$$

corroborating the conclusion reached earlier in Test # 3.

REFERENCES

1. F. D. Crary, *A versatile precompiler for nonstandard arithmetics*, ACM Trans. Mathematical Software 5 (1979), 204–217.
2. J. J. Dongarra et al., *LINPACK User's Guide*, SIAM, Philadelphia (1979).
3. W. Gautschi, *On generating orthogonal polynomials*, SIAM J. Sci. Statist. Comput. 3 (1982), 289–317.
4. R. G. Gordon, *Error bounds in equilibrium statistical mechanics*, J. Mathematical Phys. 9 (1968), 655–663.
5. Soo-Y. Lee, *The inhomogeneous Airy functions* Gi(z) *and* Hi(z), J. Chem. Phys. 72 (1980), 332–336.