Neural, Parallel, and Scientific Computations 21 (2013) 419-430

ALTERNATIVE TO GAUSS-KRONROD QUADRATURE

LAWRENCE F. SHAMPINE

Mathematics Department, Southern Methodist University
Dallas, Texas 75275 USA

ABSTRACT. Popular programs for approximating definite integrals are based on pairs of formulas. Many use a GK pair consisting of a Gauss-Legendre formula and a Kronrod extension. A simpler approach to a companion for a Gauss-Legendre formula provides a pair that costs the same, but has a significantly higher degree of precision. Like the GK pairs, the new pairs have interior nodes and positive weights. A variant of the approach applies to Gauss-Lobatto formulas.

AMS (MOS) Subject Classification. 65Y99

1. Introduction

The usual way to approximate numerically a definite integral

$$I(f) = \int_a^b f(x)dx \tag{1.1}$$

is a formula of the form

$$Q(f) = \sum_{j=1}^{N} A_j f(x_j) \tag{1.2}$$

The popular formulas have nodes x_j in $[a, b]$ and weights A_j that are positive. A useful measure of the cost of forming (1.2) is N, the number of evaluations of f. A formula has degree of precision d if it integrates exactly all polynomials of degree less than or equal to d. The degree is a measure of accuracy because if the formula has positive weights, then for *any* polynomial $p(x)$ of degree at most d,

$$|E(f)| = |I(f) - Q(f)| \leq 2(b - a)\|f - p\|_\infty.$$

It is known that a formula of the form (1.2) must have $d \leq 2N - 1$. The Gauss-Legendre formula of N nodes is popular because it has the highest possible degree of precision, the nodes all lie in (a, b) and the weights are all positive.

General-purpose quadrature programs provide an approximation to $I(f)$ that has an accuracy specified by the user. To accomplish this we need not only an approximation $Q(f)$, but also an estimate of the error of the approximation. Gauss-Legendre formulas are not so attractive when used this way. The error of $Q(f)$ is

Received July 15, 2013

conventionally estimated by comparing it to an approximation $Q^*(f)$ computed with a formula of higher degree of precision. From

$$E(f) = I(f) - Q(f) = Q^*(f) - Q(f) + E^*(f)$$

we see that the quality of the estimate

$$Q^*(f) - Q(f) \approx I(f) - Q(f)$$

depends on how small $E^*(f)$ is relative to $E(f)$, which suggests that the companion formula should be of substantially higher degree of precision. The cost of a companion formula can be reduced by sharing function evaluations with the basic formula. It would be natural to use another Gauss-Legendre formula of more nodes for $Q^*(f)$. Unfortunately, the nodes of the two formulas are all different (with the possible exception of the midpoint). To estimate efficiently the error of a Gauss-Legendre formula of M nodes with odd M, Kronrod [6] adds nodes to obtain a formula with a total of $2M+1$ nodes and a degree of precision $3M+2$. Put differently, the Gauss-Kronrod pair with a total of N nodes has degrees of precision $N-2$ and $(3N+1)/2$. The nodes and weights are obtained as solutions of nonlinear algebraic equations, so key issues are that solutions exist, the nodes are real and lie in the interval, and all weights are positive. Gautschi [5] and Notaris [10] survey theoretical work on Kronrod's approach to deriving efficient pairs of formulas. Laurie [8] surveys methods for computing Kronrod formulas and presents an effective scheme.

An alternative to Kronrod extensions was proposed by Laurie [7]. He constructs an *anti-Gaussian* formula of $M+1$ nodes that has leading error term equal in magnitude and opposite in sign to the leading error term of the Gaussian formula of M nodes. In the terminology of this paper, he uses an anti-Gaussian formula to construct a pair of N nodes with degrees $(N-2, N)$. It is to be appreciated that Laurie considers a problem that is more difficult than (1.1) because he allows a weight function in the integral. He shows that the companion formula of his approach "...has significant theoretical and practical properties. In particular, it always exists, it is an almost trivial task to construct it, it always has positive weights, its nodes are always real, and at worst two nodes may be exterior."

Computing practice has further evolved so that it is now standard practice to estimate the error in $Q(f)$, but then to use the value $Q^*(f)$. After all, the estimate of the error is not valid unless the value $Q^*(f)$ is more accurate than $Q(f)$, so why not use it? With this in mind, we have discovered a simple way to obtain pairs with degrees of precision $(N-2, 2N-1)$ when N is odd. For integrals of the form (1.1), it has the advantages that Laurie obtained in his approach and more: There is no question about the existence of the companion formula. The nodes are automatically real and lie in the interval. There is a remarkably simple expression for the weights, which are all positive.

It is as easy to implement one pair of N nodes as another. The pairs developed here are attractive because $Q(f)$ has the same degree of precision as the corresponding Gauss-Kronrod formula and $Q^*(f)$ has a substantially larger degree. To understand better how the new pairs compare to Gauss-Kronrod pairs in respect to degree of precision, we mention some specific examples. The GK15 pair developed for QUADPACK [11] consists of the 7-point Gauss-Legendre formula with a Kronrod extension for a total of 15 nodes. The formulas have degree 13 and 23, respectively. The `quadgk` [12] program of MATLAB [9] also uses this pair. Indeed, the *Guide to Available Mathematical Software* [3] shows that GK15 is in wide use. The new pair NP15 has degrees $(13, 29)$. To mention pairs of both fewer and more nodes, the `quad2d` program of MATLAB uses the 7 node scheme with degrees $(5, 11)$ and the QAGS program of QUADPACK [11] uses the 21 node scheme with degrees $(19, 32)$. The new pairs have degrees $(5, 13)$ and $(19, 41)$, respectively.

Gauss-Lobatto formulas have the highest degree of precision when both end points of the interval are required to be nodes. There has been some interest in Kronrod extensions of these formulas, a notable example being the pair of Gander and Gautschi [4] implemented in the `quadl` program of MATLAB. Though we do not pursue the matter, we do show that a variant of our approach applies to Gauss-Lobatto formulas.

It is usual to compare formulas by applying them to a collection of test problems, but there are analytical tools which can be used to compare the accuracy of formulas in a more quantitative way. We illustrate their use in Section 3 where we compare the popular GK15 pair to NP15. By every measure NP15 is at least as accurate as GK15. It is advantageous for smooth integrands and the smoother the integrand, the greater the advantage.

2. **New Pairs**

Formulas are usually developed for integration over a standard interval which we take to be $[-1, +1]$. It will be useful to review briefly interpolatory quadrature formulas. There is a unique polynomial $P_{N-1}(x)$ of degree N that interpolates $f(x)$ at N distinct nodes x_j. It is readily verified that this polynomial can be written in terms of the fundamental Lagrangian interpolating polynomials,

$$L_j(x) = \prod_{m=1, m\neq j}^{N} \left(\frac{x - x_m}{x_j - x_m} \right),$$

as

$$P_{N-1}(x) = \sum_{j=1}^{N} f(x_j) L_j(x).$$

Integration then provides a quadrature formula,

$$Q(f) = \int_{-1}^{+1} P_{N-1}(x)\, dx = \sum_{j=1}^{N} A_j f(x_j),$$

with

$$A_j = \int_{-1}^{+1} L_j(x) dx \tag{2.1}$$

By construction this formula integrates exactly any polynomial of degree $N-1$, so its degree of precision is at least $N-1$. It is the choice of nodes that gives the Gauss-Legendre formula a degree of precision that is much higher.

We consider now a Gauss-Legendre formula with $N = 2M+1$ as $Q^*(f)$. It will be convenient to use a different notation to show more clearly the symmetries of the nodes and weights. This formula can be written as

$$Q^*(f) = w_0 f(0) + \sum_{j=1}^{M} w_j \left[f(-x_j) + f(x_j) \right]. \tag{2.2}$$

In more detail, there are M positive nodes x_j, M negative nodes $-x_j$, and $x_0 = 0$. The weights corresponding to the negative nodes are equal to the weights w_j corresponding to the positive nodes. For later use we note that a formula of this form integrates exactly any odd power of x.

We construct a companion formula $Q(f)$ as the quadrature formula of degree $N-2$ based on interpolation to all the nodes of the Gauss-Legendre formula except the origin. This formula has the form (2.2) with $A_0 = 0$, so we need only compute the weights A_j with $j > 0$. This is standard, but a little trick leads to a remarkable representation of the weights. For these nodes the fundamental interpolating polynomial $L_j(x)$ is of degree $N-2$, so it is integrated exactly by the Gauss-Legendre formula of N nodes. The expression (2.2) simplifies drastically when we realize that $f(x) = L_j(x)$ has value 1 at the node x_j and vanishes at all the other nodes interpolated by the companion formula. The one node not interpolated by the companion formula is the origin, so we are left with

$$\begin{aligned} A_j &= Q^*(L_j) = w_0\, L_j(0) + w_j \\ &= w_j + \frac{w_0}{2} \prod_{m=1, m\neq j}^{M} \left(\frac{x_m^2}{x_m^2 - x_j^2} \right), \end{aligned} \tag{2.3}$$

With this result it is an easy matter to compute the coefficients as accurately as we wish with MuPAD, which is Matlab's computer algebra package. It has a function `gldata` for computing Gauss-Legendre formulas using arithmetic of any specified number of digits. And, by setting the environmental variable DIGITS appropriately, we can evaluate (2.3) accurately. For the numerical studies of Section 3 we computed the coefficients in 32 decimal arithmetic and rounded them to the 16 digits appropriate to MATLAB.

The Gauss-Lobatto formulas have the highest degree of precision possible when it is required that both end points be nodes. They have a degree of precision $2N-3$ and all $A_j > 0$. Examination of our approach to constructing a companion to Gauss-Legendre formulas shows that it applies without alteration to Gauss-Lobatto formulas of an odd number of nodes. The companion again has degree of precision $N-2$. Gander and Gautschi [4] derive a Kronrod extension with positive weights for the Lobatto formula of 4 points. The pair has a total of 7 points and degrees of precision $(5,9)$. This pair is implemented in the MATLAB program `quadl`. The new approach leads to a pair with degrees $(5,11)$.

For small N the Gauss-Legendre formulas can be evaluated analytically and so can the NP formulas. Table 1 provides the coefficients for $N=5$. The same is true for the Gauss-Lobatto formulas, so we display in Table 2 the coefficients for $N=5$. The negative coefficients of this companion formula do not make it uninteresting. One reason is that the formula actually used for the integration does have positive coefficients. Another is that the companion is used only for error estimation and negative coefficients are important then only if they are large in magnitude.

TABLE 1. NP5–values shown for non-negative nodes.

nodes	0	$\frac{\sqrt{5-2\sqrt{10/7}}}{3}$	$\frac{\sqrt{5+2\sqrt{10/7}}}{3}$
d = 3	0	$\frac{1}{2}+\frac{\sqrt{70}}{20}$	$\frac{1}{2}-\frac{\sqrt{70}}{20}$
d = 9	$\frac{128}{225}$	$\frac{322+13\sqrt{70}}{900}$	$\frac{322-13\sqrt{70}}{900}$

TABLE 2. NPL5–values shown for non-negative nodes.

nodes	0	$\sqrt{\frac{3}{7}}$	1
d = 3	0	$\frac{7}{6}$	$-\frac{1}{6}$
d = 3	$\frac{4}{9}$	$\frac{7}{9}$	0
d = 7	$\frac{32}{45}$	$\frac{49}{90}$	$\frac{1}{10}$

There is a variation on our approach to a companion for Gauss-Lobatto formulas which seems always to produce formulas with positive coefficients, so we favor it. The idea is to drop the two end points rather than the midpoint. The equivalent of (2.3) then has two forms:

$$\begin{aligned} A_0 &= w_0 + w_M \left[L_0(-1) + L_0(+1)\right] \\ &= w_0 + 2\, w_M \prod_{m=1}^{M-1} \left(\frac{1 - x_m^2}{0 - x_m^2} \right), \end{aligned} \tag{2.4}$$

and for $j > 0$,

$$A_j = w_j + \frac{w_M}{x_j^2} \prod_{m=1, m \neq j}^{M-1} \left(\frac{1 - x_m^2}{x_j^2 - x_m^2} \right). \tag{2.5}$$

We show this formula for $N = 5$ in Table 2 as the second formula with degree of precision 3. Dropping two nodes leads to an interpolating polynomial of even degree $N - 3 = 2M - 2$. However, a formula of the form (2.2) integrates exactly odd powers, so this formula actually has degree of precision $N - 2$. It is pleasant that it has the same degree of precision as the one obtained by dropping only one point and the weights are all positive.

It is known that the Gauss-Legendre formulas must have positive weights and the Kronrod extensions are computed so as to have positive weights. An interesting theoretical question is whether the companion formulas of our approach must have positive weights. The *practical* issue of companion formulas having positive weights is settled by direct computation—all the companion formulas that we have computed have weights that are clearly positive. In particular, that is true of the formulas with $N = 3, 5, \ldots, 61$. Indeed, the coefficients do not vary greatly in magnitude. When we constructed each formula, we computed the ratio of the maximum weight to the minimum weight. Considering all the formulas, the largest ratio was about +769. We used a MATLAB program of von Winckel [14] and the expressions (2.4, 2.5) to compute the coefficients of the companion obtained by dropping the end points of the Gauss-Lobatto formula. The largest ratio of the maximum to minimum coefficient over the formulas $N = 3, 5, \ldots, 61$ was about +14, showing that the weights are positive and of comparable size.

3. **NP15 vs GK15**

In this section we compare the popular GK15 to the new pair, NP15, using a wide variety of analytical techniques. Proper interpretation of the results requires some discussion of the way the pair is to be used. We have in mind an adaptive implementation, which has proved very popular in software for quadrature. In broad terms an adaptive implementation approximates integrals of $f(x)$ over subintervals of $[a, b]$. If the approximation is not sufficiently accurate on a particular subinterval $[\alpha, \beta]$, the program replaces it with subintervals $[\alpha, (\alpha+\beta)/2]$ and $[(\alpha+\beta)/2, \beta]$. When the program has sufficiently accurate approximate integrals on all subintervals of $[a, b]$, the results are summed to get an approximation to (1.1). In practice $f(x)$ is often piecewise smooth. For such an integrand, many of the integrals arising in an adaptive implementation are over subintervals where the integrand is analytic. However, there are also a few subintervals where the integrand has low differentiability. Because of this it is important to understand how the formulas perform when the integrand is not smooth, while recognizing that often the integrand *is* smooth.

The classical way of expressing the error of a formula is

$$E(f) = c\left(\frac{b-a}{2}\right)^{d+2} f^{(d+1)}(\xi) \tag{3.1}$$

Here d is the degree of precision, the constant c is characteristic of the formula, and ξ is an unknown point in $[a, b]$. Using this expression we can compare the formulas with degree 13 of the GK15 and NP15 pairs. The error constant of the new formula is smaller by a factor of about 0.71.

The error expression (3.1) allows us to compare formulas only if they have the same degree of precision and only if $f(x) \in C^{(d+1)}[a, b]$. A result of [1, §4.8] based on approximation theory states that if the formula has degree d and all $A_j > 0$, then for any $f \in C^{r+1}[a, b]$ with $r \leq d$,

$$|E(f)| \leq \frac{c_r}{d^{r+1}}\left(\frac{b-a}{2}\right)^{r+2} \|f^{(r+1)}\|_\infty \tag{3.2}$$

The constant c_r here does *not* depend on the formula. Table 3 compares the two formulas of higher degree of precision by the ratio of the two bounds for integrands of modest differentiability. We see that even for $f(x)$ of low differentiability, the higher degree of precision of the NP15 pair is advantageous.

Table 3. Error bound of NP15 compared to error bound of GK15 for $f \in C^{r+1}$.

r	0	1	2	3	4	5	6	7	8
ratio	0.79	0.63	0.50	0.40	0.31	0.25	0.20	0.16	0.12

The bound (3.2) provides valuable insight when comparing formulas, but it does not distinguish formulas of the same degree of precision. Also, it does not provide information about how formulas behave for smoother integrands. We now follow [2, 13] in using norms of error functionals to compare *specific* formulas when applied to functions with smoothness that range from low differentiability to analytic. We conclude with an appraisal of the two pairs.

3.1. **Error in $C^{r+1}[a,b]$.** For $f \in C^{r+1}[a,b]$, it is well-known that the error of a formula of the form (1.2) that has degree of precision $d \geq r$ can be written as

$$I(f) - Q(f) = E(f) = \int_a^b f^{(r+1)}(t)\, K_r(t)\, dt. \tag{3.3}$$

Using the notation

$$(x-t)_+^r = \begin{cases} (x-t)^r & \text{when } x \geq t, \\ 0 & \text{when } x < t, \end{cases}$$

the Peano kernel $K_r(t)$ can be written as

$$K_r(t) = \frac{(b-t)^{r+1}}{(r+1)!} - \frac{1}{r!} \sum_{j=1}^{N} A_j\, (x_j - t)_+^r. \tag{3.4}$$

As in [13], the representation (3.3) leads to the error bound

$$|E(f)| \leq \left(\int_{-1}^{+1} K_r^2(t)\, dt \right)^{1/2} \left(\frac{b-a}{2} \right)^{r+2} \|f^{(r+1)}\|_2. \tag{3.5}$$

Other norms are commonly used in this situation, but as pointed out in [13], it is possible to evaluate exactly this particular norm of the Peano kernel. Indeed, we developed for [13] a MATLAB program that computes $\|K_r\|_2$ for given $r \geq 0$ for any formula of the form (1.2) with interval $[-1,1]$. Here we have used the program to compute norms for the GK15 and NP15 formulas for a range of r. The results are displayed in Fig. 1. Though the formulas of NP15 are more accurate in this measure than the formulas of GK15, there is little difference for integrands of low differentiability.

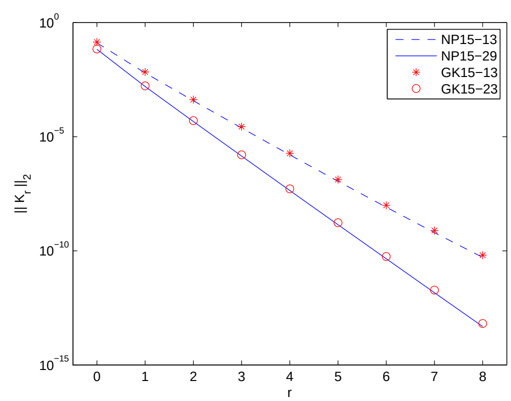

FIGURE 1. $\|K_r\|_2$ for $f \in C^{r+1}$.

3.2. **Error in $L^2(\mathcal{E}_\rho)$.** Davis and Rabinowitz [2] study the error of quadrature formulas for integrands that are analytic on the interval of integration. It is assumed that the integrand is a regular analytic function in an ellipse $\mathcal{E}_\rho$ that is defined in terms of a parameter ρ. This ellipse has its major axis along the x axis and its foci at $(-1,0)$ and $(1,0)$. The semimajor axis $a = 0.5(\rho^{1/2} + \rho^{-1/2})$ and semiminor axis $b = 0.5(\rho^{1/2} - \rho^{-1/2})$. Davis and Rabinowitz develop a bound for the error of a quadrature formula with integrand $f \in L^2(\mathcal{E}_\rho)$,

$$|E(f)| \leq \sigma \|f\|$$

The norm σ of the error functional can be expressed in terms of the Chebyshev polynomials of the second kind, which for $-1 \leq x \leq +1$ can be written as

$$U_n(x) = \frac{\sin\left((n+1)\arccos(x)\right)}{\sin(\arccos(x))}$$

Davis and Rabinowitz show that if $\tau_n = 0$ for odd n and $\tau_n = 2/(n+1)$ for even n, then

$$\sigma^2 = \frac{4}{\pi} \sum_{n=0}^{\infty} \frac{n+1}{\rho^{n+1} - \rho^{-n-1}} \left[\tau_n - \sum_{j=1}^{N} A_j \, U_n(x_j) \right]^2 \tag{3.6}$$

In this class of integrands the parameter ρ measures how far the nearest singularity is from the interval of integration. However, in the spirit of [2], we plot σ against the semimajor axis a in Fig. 2 for $1.01 \leq a \leq 1.50$. We see that the new pair is more accurate, but there is little difference in the accuracy of the formulas of degree 13. The value $a = 1.01$ at one end of the range permits a singularity that is quite close to the interval of integration. The new formula of higher degree offers very little advantage then, but it is increasingly advantageous for smoother integrands.

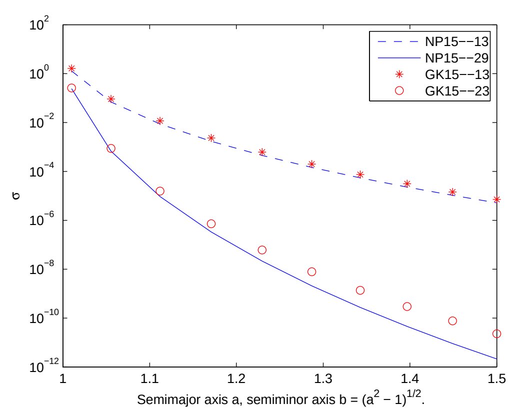

FIGURE 2. Error norm for $f \in L^2(\mathcal{E}_\rho)$ with $\rho = (a+b)^2$.

The largest value of a in Fig. 2 corresponds to $\rho \approx 6.8541$. To compare the formulas for integrands smoother than this, we can use the first non-zero terms in (3.6). In this connection we note that all the formulas considered have the form (2.2) and as pointed out in Section 2, they integrate exactly all odd powers of x. The $U_n(x)$ consist of odd powers of x when n is odd, so all terms in (3.6) with n odd are zero. For this reason the first non-zero term is a useful approximation for ρ of even modest size.

Comparing the first non-zero terms is especially simple when the formulas have the same degree of precision because ρ does not appear in the ratio. A little calculation shows that the ratio of the norm for the formula of degree 13 of NP15 to the norm of the formula of degree 13 of GK15 is about 0.725. In this measure we see that the new formula is somewhat more accurate for integrands that are quite smooth. The ratio for the formulas of higher degree of precision is complicated by the presence of ρ, but the leading terms have n that are rather large and we have in mind ρ at least as big as 6.8541, so we can approximate $\rho^{n+1} - 1/\rho^{n+1} \approx \rho^{n+1}$. With this, the ratio of the norms for the two higher order formulas is about $35.8/\rho^3$. This shows that the new formula is substantially more accurate than the Kronrod formula for smooth integrands and the smoother the integrand, the greater the advantage.

3.3. **An Appraisal.** The two pairs have the same number of nodes and both have formulas with weights of one sign. It is as easy to implement one as the other. By every measure of accuracy considered, NP15 is at least as accurate as GK15. Standard practice is to control the error of the formula of lower degree, but actually use the formula of higher degree. The two formulas of lower degree have roughly the same accuracy. It is common that the integrand is piecewise smooth, in which case an adaptive implementation approximates the integrand over many subintervals where it is smooth. The analysis of Section 3.2 shows that the formula of higher degree in NP15 can be much more accurate than the corresponding formula of GK15 when the integrand is smooth. These observations show that the error estimate of NP15 is better and the control of error is more conservative than with GK15 because a more accurate result is used. If one were to write a new quadrature program, there seems to be no reason not to choose NP15 instead of GK15. The matter is less clear when we ask whether we should change the pair in a piece of software that is already in wide use because the decision involves matters other than the relative merits of the pairs.

4. **Conclusions**

Software based on Gauss-Kronrod pairs in an adaptive implementation are very popular for the approximation of definite integrals. A Gauss-Kronrod pair of N nodes has degrees $(N-2, N)$. The corresponding pair derived here for odd N has degrees $(N-2, 2N-1)$. Because of the higher degree of precision, the new pair has a more accurate error estimate and a more conservative control of error. Both pairs have the desirable properties of interior nodes and positive weights. There seems to be no reason not to prefer the new pair when writing software for definite integrals.

REFERENCES

[1] P. J. Davis and P. Rabinowitz, *Methods of Numerical Integration*, 2nd ed., Academic, Orlando, 1984.

[2] P. J. Davis and P. Rabinowitz, On the Estimation of Quadrature Errors for Analytic Functions, *Math. Comp. (MTAC)*, 8:193–203, 1954.

[3] *GAMS: The Guide to Available Mathematical Software*, `http://gams.nist.gov/`.

[4] W. Gander and W. Gautschi, Adaptive Quadrature–Revisited, *BIT*, 40:84–101, 2000.

[5] W. Gautschi, Gauss-Kronrod Quadrature — a Survey, pp. 39–66 in *Numerical Methods and Approximation Theory IV*, G.V. Milanović, ed., Univ. Niš, Niš, 1988.

[6] A. Kronrod, *Nodes and Weights of Quadrature Formulas*, Consultants Bureau, New York, 1965.

[7] D. P. Laurie, Anti-Gaussian Quadrature Formulas, *Math. Comp.*, 65:739–747, 1996.

[8] D. P. Laurie, Calculation of Gauss-Kronrod Quadrature Rules, *Math. Comp.*, 66:1133–1145, 1997.

[9] MATLAB, The MathWorks, Inc., 3 Apple Hill Dr., Natick, MA 01760.

[10] S. Notaris, An Overview of Results on the Existence or Nonexistence and the Error Term of Gauss-Kronrod Quadrature Formulae, pp. 485–496 in *Approximation and Computation: A Festschrift in Honor of Walter Gautschi*, R. V. M. Zahar, ed., Internat. Ser. Numer. Math., v. 119, Birkhäuser, Boston, 1994.

[11] R. Piessens, E. de Doncker-Kapenga, C. Überhuber, and D. Kahaner, *QUADPACK: Subroutine Package for Automatic Integration*, Springer Verlag, New York, 1983.

[12] L. F. Shampine, Vectorized Adaptive Quadrature in MATLAB, *J. Comp. Appl. Math.*, 211:131–140, 2008.

[13] L. F. Shampine, Quadrature Error Estimates for Integrands with Modest Differentiability, *Canadian Appl. Math. Quarterly*, 17:571–583, 2009.

[14] G. von Winckel, `lglnodes`: Legendre-Gauss-Lobatto Nodes and Weights, `http://www.mathworks.com/matlabcentral/fileexchange/`.