

RELIABLE AUTOMATIC QUADRATURE FOR CONES OF INTEGRANDS

FRED J. HICKERNELL, MARTHA RAZO, AND SUNNY YUN

ABSTRACT. Hi

1. INTRODUCTION

Automatic numerical algorithms for univariate integration appear in many widely used software packages, such as MATLAB (2013), Mathematica (2012), NAG (2012), and R (2012). Automatic algorithms determine adaptively where and how much to sample the integrand by estimating the algorithm error from the sampled function values. Unfortunately, such error estimation methods are unreliable, as pointed out by James Lyness (1983). We now know that Lyness’s objections can be overcome by employing a new paradigm for error estimation for cones of integrands. The discussion here illustrates with greater detail the more general setting in Clancy et al. (2013).

Soon after a calculus student first learns about integration, she learns that the integrals of many elementary functions are *not* themselves elementary functions. An important example is the standard normal probability density function, $\phi : x \mapsto e^{-x^2}/\sqrt{2\pi}$. Since the antiderivative of ϕ cannot be expressed in terms of elementary functions, we need tables in textbooks and special functions in calculators to provide the values of the standard normal distribution function $\Phi : x \mapsto \int_{-\infty}^x \phi(t) dt$. By contrast, the derivatives of elementary functions are elementary functions.

1.1. The Trapezoidal Rule. Students need to know that the lack of an analytic answer does not mean that no answer exists, so calculus students are often introduced to numerical integration (quadrature) methods such as the trapezoidal rule, T_n :

$$(1) \quad \int_0^1 f(x) dx \approx T_n(f) := \frac{1}{2n} [f(0) + 2f(1/n) + \cdots + 2f(1 - 1/n) + f(1)].$$

It is then natural to ask, “How large should the number of trapezoids, n , be so that the absolute error of $T_n(f)$ does not exceed a tolerance ε ?” There is an error bound of the form

$$(2) \quad \text{err}(f, n) := \left| \int_0^1 f(x) dx - T_n(f) \right| \leq \frac{V(f')}{8n^2}, \quad n \in \mathbb{N} = \{1, 2, \dots\}.$$

Here, $V(\cdot)$ denotes the total variation, which is defined as

$$(3) \quad V(f) := \sup_{\substack{n \in \mathbb{N} \\ 0=x_0 < x_1 < \cdots < x_n=1}} \sum_{i=1}^n |f(x_i) - f(x_{i-1})|.$$

The norm $V(\cdot)$ is related to \mathcal{L}_p -norms as follows:

$$(4) \quad V(f) = \|f'\|_1 \text{ if } \|f'\|_1 < \infty, \quad \|f\|_p := \begin{cases} \left[\int_0^1 |f(x)|^p dx \right]^{1/p}, & 1 \leq p < \infty, \\ \sup_{0 \leq x \leq 1} |f'(x)|, & p = \infty. \end{cases}$$

Error bounds resembling (2) exist in terms of \mathcal{L}_p -norms of f'' but with different leading constants.

If f has a simple form, one might be able to derive an upper bound, M_1 , on this norm, i.e., $V(f') \leq M_1$. In this case choosing any $n \geq \sqrt{M_1/(8\varepsilon)}$ ensures that the error tolerance is met. For example, for ϕ defined above, $V(f') = 1/\sqrt{2\pi e}$, and so one may choose any $n \geq 1/\sqrt{8\sqrt{2\pi e}\varepsilon}$.

2. THE FLAWS IN THE PREVALENT DATA-BASED ERROR ESTIMATE

It is often impractical to bound $V(f')$ analytically because the formula for f is not given explicitly or because it is too complicated. A more widely applicable quadrature method would reliably estimate the error based solely on the function values, $f(x_i)$, and determine the sample size appropriately. Numerical methods texts, such as Burden and Faires (2010, p. 223–224), Cheney and Kincaid (2013, p. 233), and Sauer (2012, p. 270), advise the readers to estimate the error by comparing two trapezoidal rules with different numbers of trapezoids, specifically,

$$(5) \quad \widehat{\text{err}}(f, n) = \frac{T_n(f) - T_{n/2}(f)}{3}, \quad \frac{n}{2} \in \mathbb{N}.$$

One may then compute $\widehat{\text{err}}(f, 2^j)$ for $j = 1, 2, \dots$ and stop when $\widehat{\text{err}}(f, 2^j) \leq \varepsilon$. By doubling the number of trapezoids for each iteration, the data from the previous iterations can be reused.

Unfortunately, it is easy to construct integrands, f , such that this error estimate fails completely, e.g.,

$$(6) \quad \int_0^1 f(x) dx = 1, \quad T_n(f) = T_{n/2}(f) = -1, \quad \text{err}(f, n) = 2 \neq 0 = \widehat{\text{err}}(f, n).$$

Figure 1 shows two examples satisfying these conditions. These two examples illustrate the two ways in which the error estimate $\widehat{\text{err}}(\cdot)$, and any automatic quadrature algorithm base on it, can fail. One way is ultimately unavoidable but ought to be quantifiable. The other way is inexcusable, but is prevalent in widely used automatic numerical quadrature.

2.1. Spiky Integrands. Any quadrature algorithm that depends on just function values to estimate error will be fooled by a spiky integrands. Suppose that the quadrature algorithm samples the integrand at a sequence of points, ξ_1, ξ_2, \dots , and suppose that $f(\xi_i)$ is the same value for all ξ_i . Here the algorithm may even be adaptive, meaning that ξ_{i+1} may depend on $(\xi_1, f(\xi_1)), \dots, (\xi_i, f(\xi_i))$. Eventually the algorithm stops at n points. Under mild conditions on the set of integrands of interest the best estimate of the integral is that constant.

Figure 1 (left) plots a spiky integrand that is defined in terms of the quartic Bernoulli polynomial:

$$(7) \quad f_{\text{spiky}}(x; n) = 1 + 60B_4(\{nx\}), \quad n \in \mathbb{N}, \quad \{x\} := x \bmod 1.$$

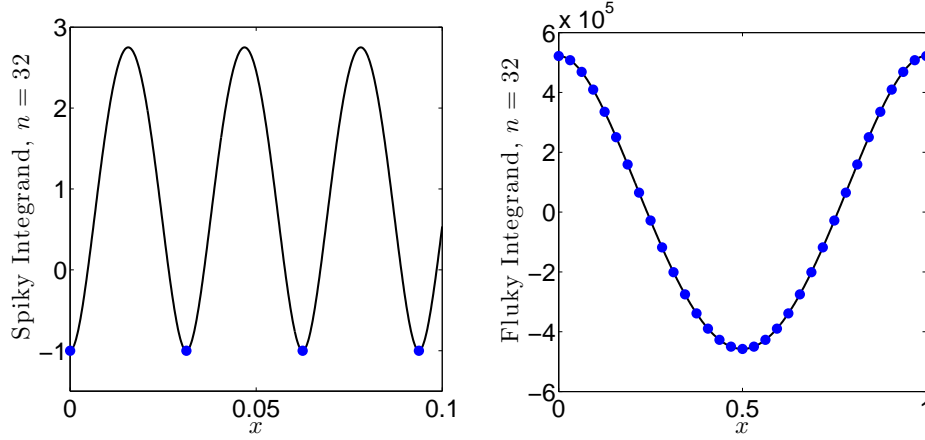


FIGURE 1. Examples of a spiky integrand (left) and a fluky integrand (right), which satisfy (6) for $n = 32$.

The Bernoulli polynomials are described by Abramowitz and Stegun (1964, Chap. 23, <http://dlmf.nist.gov/24>). The first six are defined as follows

$$(8a) \quad B_0(x) = 1, \quad B_1(x) = x - 1/2, \quad B_2(x) = 1/6 - x(1 - x),$$

$$(8b) \quad B_3(x) = \frac{1}{2}x(1 - x)(1 - 2x), \quad B_4(x) = -1/30 + [x(1 - x)]^2,$$

$$(8c) \quad B_5(x) = \frac{1}{6}x(1 - x)(1 - 2x)(1 + 3x - 3x^2).$$

The integrand $f_{\text{spiky}}(\cdot; n)$ satisfies (6) for even n and has

- $n + 1$ local minima of -1 at $x = 0, 1/n, \dots, 1$, and
- n local maxima of $11/4$ at $x = 1/(2n), 3/(2n), \dots, 1 - 1/(2n)$.

The problem of integrating spiky functions never goes away, but it should be quantifiable. An algorithm ought to integrate functions accurately, provided that they have spikes no thinner than say, δ . Moreover, the algorithm should be tunable so that the user can specify δ . We describe such an algorithm in Section 3, along with a precise definition of spike width.

2.2. Fluky Integrands. Thirty years ago James Lyness (1983, p. 69) made the following observation in his SIAM Review article, *When Not to Use an Automatic Quadrature Routine*:

While prepared to take the risk of being misled by chance alignment of zeros in the integrand function, or by narrow peaks which are “missed,” the user may wish to be reassured that for “reasonable” integrand functions which do not have these characteristics all will be well. It is the purpose of the rest of this section to demonstrate by example that he cannot be reassured on this point. In fact the routine is likely to be unreliable in a significant proportion of the problems it faces (say 1 to 5%) and there is no way of predicting in a straightforward way in which of any set of apparently reasonable problems this will happen.

Lyness went on to describe how to construct, what we would call a fluky integrand.

Figure 1 (right) plots a fluky integrand, which is defined in terms of the quadratic and quartic Bernoulli polynomials:

$$(9) \quad f_{\text{fluky}}(x; n) = 1 - 15n^2[B_2(x) + n^2B_4(x)], \quad n \in \mathbb{N}.$$

The integrand $f_{\text{fluky}}(\cdot; n)$ satisfies (6) for even n and has only

- 1 local minimum of $(16 + 20n^2 - 7n^4)/16$ at $x = 1/2$, and
- 2 local maxima of $(19 - 10n^2 + 2n^4)/4$ at $x = (1 \pm \sqrt{1 - 2/n^2})/2$.

Integrands like this one are termed by us as fluky because their construction is rather delicate. They satisfy conditions (6) without a large number of local optima. To illustrate this delicacy, note that removing either the B_2 or the B_4 term does not affect the value of the integral but does cause the trapezoidal rule estimates of the integral to vary significantly for n and $n/2$ trapezoids.

The failure of the error estimate in (5) for $f_{\text{fluky}}(\cdot; n)$ seems avoidable. For even moderate n , the sampled function values, $(f_{\text{fluky}}(i/n; n))_{i=1}^n$, are spread across a large range. This suggests that true error, $\text{err}(f_{\text{fluky}}(\cdot; n), n)$, might be substantial, and far from the error estimate, $\widehat{\text{err}}(f_{\text{fluky}}(\cdot; n), n) = 0$. In Section 3 we transform this intuition into a better method for estimating the error of the trapezoidal rule.

2.3. Necessary Conditions for Failure of the Error Estimate. Integrands that satisfy (6) and fool the error estimate in (5) must satisfy certain conditions. First note that the error bound in (2)

$$V(f') \geq 8n^2 \text{err}(f, n) = 16n^2,$$

so the total variation of the first derivative must be increasingly large as the number of trapezoids increases. Moreover, the trapezoidal rule added to the error estimate in (2) results in Simpson's rule:

$$\begin{aligned} S_n(f) &:= T_n(f) + \widehat{\text{err}}(f, n) = \frac{4T_n(f) - T_{n/2}(f)}{3} \\ &= \frac{1}{3n} [f(0) + 4f(1/n) + 2f(2/n) + 4f(3/n) + \cdots + 4f(1 - 1/n) + f(1)] \end{aligned}$$

The error bound of Simpson's rule then serves as an bound on the error of the error estimate:

$$(10) \quad |\text{err}(f, n) - \widehat{\text{err}}(f, n)| = \left| \int_0^1 f(x) dx - S_n(f) \right| \leq \frac{V(f''')}{???n^4}, \quad \frac{n}{2} \in \mathbb{N}.$$

So, integrands satisfying (6) must also satisfy

$$V(f''') \geq ??n^4 |\text{err}(f, n) - \widehat{\text{err}}(f, n)| = ???n^4.$$

Thus, the total variation of the third derivative must be increase even faster as the number of trapezoids increases.

The derivatives of Bernoulli polynomials are multiples of lower degree Bernoulli polynomials and the integrals of nonconstant Bernoulli polynomials vanish (Abramowitz and Stegun, 1964, Chap. 23, <http://dlmf.nist.gov/24>):

$$B'_n(x) = nB_{n-1}(x), \quad \int_0^1 B_{n+1}(x) dx, \quad n \in \mathbb{N}.$$

The formula for the derivatives facilitates the computation of the total variation of the spiky and fluky examples above. For the spiky integrand defined in (7),

$$(11a) \quad f'_{\text{spiky}}(x; n) = 240nB_3(\{nx\}), \quad f''_{\text{spiky}}(x; n) = 720n^2B_2(\{nx\}),$$

$$(11b) \quad f'''_{\text{spiky}}(x; n) = 1440n^3B_1(\{nx\}),$$

$$(11c) \quad V(f_{\text{spiky}}(\cdot; n)) = \frac{15n}{2}, \quad V(f'_{\text{spiky}}(\cdot; n)) = \frac{80n^2}{\sqrt{3}} > 16n^2,$$

$$(11d) \quad V(f''_{\text{spiky}}(\cdot; n)) = 360n^3, \quad V(f'''_{\text{spiky}}(\cdot; n)) = 1440n^4.$$

For the fluky integrand defined in (9),

$$(12a) \quad f'_{\text{fluky}}(x; n) = -30n^2[B_1(x) + 2n^2B_3(x)],$$

$$(12b) \quad f''_{\text{fluky}}(x; n) = -30n^2[1 + 6n^2B_2(x)], \quad f'''_{\text{fluky}}(x; n) = -360n^4B_1(x),$$

$$(12c) \quad V(f_{\text{fluky}}(\cdot; n)) = \frac{15}{8}(8 - 4n^2 + n^4),$$

$$(12d) \quad V(f'_{\text{fluky}}(\cdot; n)) = \frac{10n}{3} \left[9n + 2\sqrt{3(-2 + n^2)^3} \right] > 16n^2,$$

$$(12e) \quad V(f''_{\text{fluky}}(\cdot; n)) = 90n^4, \quad V(f'''_{\text{fluky}}(\cdot; n)) = 360n^4.$$

Note that the variations of the integrand its derivatives behaves differently for these two examples:

$$V(f_{\text{spiky}}^{(j)}(\cdot; n)) \asymp n^{j+1}, \quad V(f_{\text{fluky}}^{(j)}(\cdot; n)) \asymp n^4, \quad j = 0, \dots, 3.$$

For the spiky integrand the asymptotic order of the total variation for large n increases with the order of the derivative while for the fluky integrand it stays the same.

3. A GUARANTEED QUADRATURE ALGORITHM

Now we turn to a new error estimate for numerical integration proposed in

4. ACKNOWLEDGEMENTS

The authors are grateful for discussions with a number of colleagues. This research is supported in part by grant NSF-DMS-1115392.

REFERENCES

- Abramowitz, M. and I. A. Stegun (eds.) 1964. *Handbook of mathematical functions with formulas, graphs and mathematical tables*, U. S. Government Printing Office, Washington, DC.
- Burden, R. L. and J. D. Faires. 2010. *Numerical analysis*, Ninth, Cengage Brooks/Cole, Belmont, CA.
- Cheney, W. and D. Kincaid. 2013. *Numerical mathematics and computing*, Seventh, Brooks/Cole, Boston.
- Clancy, N., Y. Ding, C. Hamilton, F. J. Hickernell, and Y. Zhang. 2013. *The complexity of guaranteed automatic algorithms: Cones, not balls*. submitted for publication, arXiv.org:1303.2412 [math.NA].
- Lyness, J. N. 1983. *When not to use an automatic quadrature routine*, SIAM Rev. **25**, 63–87.
- R Development Core Team. 2012. *The R Project for Statistical Computing*.
- Sauer, T. 2012. *Numerical analysis*, Pearson.
- The MathWorks, Inc. 2013. *MATLAB 8.1*, Natick, MA.
- The Numerical Algorithms Group. 2012. *The NAG library*, Mark 23, Oxford.
- Wolfram Research Inc. 2012. *Mathematica 9*, Champaign, IL.