Tanh-Sinh High-Precision Quadrature

David H. Bailey[1]

19 Jan 2006

1. Introduction

In the past few years, computation of definite integrals to high precision has emerged as a very useful tool in experimental mathematics. In particular, it is often possible to recognize an otherwise unknown definite integral in analytic terms, provided its numerical value can be calculated to high accuracy. As a single example, recently the author, together with Jonathan Borwein and Richard Crandall, were able to evaluate the integrals

$$C_n = \frac{4}{n!}\int_0^\infty \cdots \int_0^\infty \frac{1}{\left(\sum_{j=1}^n (u_j + 1/u_j)\right)^2} \frac{du_1}{u_1} \cdots \frac{du_n}{u_n},$$

for n as large as 1024, to 1000-digit precision. This was done by first transforming them into the 1-D integrals

$$C_n = \frac{2^n}{n!}\int_0^\infty t K_0^n(t)\, dt,$$

where $K_0(t)$ denotes the modified Bessel function, and then applying the tanh-sinh quadrature scheme. One can show that $C_1 = 2, C_2 = 1, C_3 = L_{-3}(2) = \sum_{n \geq 0}[1/(3n+1)^2 - 1/(3n+2)^2]$, and the numerical values agree with these. Using the high-precision value of C_4, it was then conjectured that $C_4 = 7\zeta(3)/12$. We also discovered that $\lim_{n\to\infty} C_n = 2e^{-2\gamma}$, where γ is Euler's constant. These experimental discoveries have subsequently been proven [3].

The tanh-sinh quadrature scheme is the fastest currently known high-precision quadrature scheme, particularly when one counts the time for computing abscissas and weights [4]. It has been successfully employed for quadrature calculations up to 20,000-digit precision [2]. It works well for functions with blow-up singularities or infinite derivatives at endpoints [4], and it is well-suited for highly parallel implementation [2]. Rigorous error bounds can be easily computed [1]. It was first introduced by Takahasi and Mori [5].

2. The Basic Tanh-Sinh Quadrature Scheme

The tanh-sinh quadrature scheme is based on the Euler-Maclaurin summation formula, which implies that for certain bell-shaped integrands, approximating the integral by a simple step-function summation is remarkably accurate. This principle is utilized in the tanh-sinh scheme by transforming the integral of $f(x)$ on the interval $[-1, 1]$ to an integral on $(-\infty, \infty)$ using the change of variable $x = g(t)$, where $g(t) = \tanh(\pi/2 \cdot \sinh t)$. Note that $g(x)$ has the property that $g(x) \to 1$ as $x \to \infty$ and $g(x) \to -1$ as $x \to -\infty$, and

[1]Lawrence Berkeley National Laboratory, Berkeley, CA 94720 dhbailey@lbl.gov. This work was supported in part by the Director, Office of Computational and Technology Research, Division of Mathematical, Information, and Computational Sciences of the U.S. Department of Energy, under contract number DE-AC03-76SF00098.

all derivatives rapidly approach zero for large positive and negative arguments. Thus one can write, for $h > 0$,

$$\int_{-1}^{1} f(x)\,dx \;=\; \int_{-\infty}^{\infty} f(g(t))g'(t)\,dt \;\approx\; h \sum_{j=-N}^{N} w_j f(x_j), \tag{1}$$

where $x_j = g(hj)$ and $w_j = g'(hj)$, and where N is chosen large enough that $|w_j f(x_j)| < \epsilon$ for $|j| > N$. Here $\epsilon = 10^{-p}$, where p is the numeric precision level in digits. Because $g'(t)$ and higher derivatives tend to zero very rapidly for large t, the resulting integrand $f(g(t))g'(t)$ typically is a smooth bell-shaped function for which the Euler-Maclaurin formula applies, even in cases where $f(x)$ has an infinite derivative or a blow-up singularity at one or both endpoints. As a result, the approximation in (1) is often very accurate.

Formula (1) is the gist of the tanh-sinh scheme. Note that the abscissas x_j and the weights w_j can be computed once for a given h, and then used for numerous problems. Typically one selects $h = 2^{-m}$, for some m. The author has found that $m = 12$ is more than sufficient to evaluate most integrals to 1000-digit accuracy. One typically proceeds one "level" at a time, where level k uses $h = 2^{-k}$, starting with level one and continuing until either a fully accurate result has been obtained (see Section 4) or the final (m-th) level has been completed.

In a straightforward implementation, with p-digit arithmetic, abscissa-weight pairs are computed with relative precision p digits and, for a given h, are generated for $j \geq 0$ until $w_j < 10^{-p}$. Note that $x_{-j} = -x_j$ and $w_{-j} = w_j$, so x_j and w_j do not need to be computed for $j < 0$. Also note that once the array of abscissas and weights has been calculated for $h = 2^{-m}$, then the abscissas and weights for any level $k \leq m$ may be found by simply skipping through this array with a stride of 2^{m-k}. Also note that the integrand function needs to be evaluated only at the odd-indexed abscissas at each level (after the first level), since the sum of the function-weight products at the even-indexed abscissas has already been computed—this is merely the quadrature result from the previous level. In this way, considerable run time can be saved.

3. Enhancements to the Basic Scheme

In a more sophisticated implementation, one can calculate additional abscissa-weight pairs, for a given h and precision level p, continuing until $w_j < 10^{-np}$ for some $n \geq 2$ (the author normally uses $n = 2$). Also, it is better to store $y_j = 1 - x_j$ instead of x_j, since x_j are very close to one for large j. The y_j can be computed more accurately as $y_j = 1/(e^{\pi/2 \cdot \sinh hj} \cosh(\pi/2 \cdot \sinh hj))$. Then during a quadrature calculation, one uses np-digit precision to calculate the abscissas x_j as $x_j = 1 - y_j$. Also, in cases where the integration interval is $[a, b]$, instead of $[-1, 1]$, one must linearly scale the argument to $[-1, 1]$ using np-digit arithmetic. With these changes, the argument values for an expression such as $1 - t$ appearing in an integrand such as $\int_0^1 e^t (1-t)^{-1/2}\,dt$ (which has a blow-up singularity at $t = 1$) are more accurate. The function itself does not need to be computed using this higher precision, so the added computational cost of this np-digit precision scaling procedure is negligible. These modifications permit one to obtain full p-digit precision in most cases where the function is badly behaved at the endpoints.

4. Error Estimation

A highly accurate estimate of the error in tanh-sinh quadrature is given by

$$\hat{E}(h) \;=\; h\left(\frac{h}{2\pi}\right)^2 \sum_{j=-N}^{N} F''(jh),$$

where $F(t) = f(g(t))g'(t)$ and $g(t) = \tanh(\pi/2 \cdot \sinh t)$ [1]. In less formal usage, one can employ the following heuristic error estimation scheme, which is inspired by the quadratically convergent behavior often achieved by the tanh-sinh scheme. Let S_k be the computed approximation of the integral for level k. Then the estimated error at level n is one if $n \le 2$, zero if $S_n = S_{n-1}$, and otherwise 10^d, where $d = \max(d_1^2/d_2,\ 2d_1,\ d_3,\ d_4)$ (except d is not set greater than 0). In this formula, $d_1 = \log_{10}|S_n - S_{n-1}|$, $d_2 = \log_{10}|S_n - S_{n-2}|$, $d_3 = \log_{10}(\epsilon \cdot \max_j |w_j f(x_j)|)$, and $d_4 = \log_{10}\max(|w_l f(x_l)|, |w_r f(x_r)|$. Here x_l is the closest abscissa to the left endpoint, x_r is the closest abscissa to the right endpoint, and $\epsilon = 10^{-p}$. Calculations of d may be done to ordinary double precision accuracy (i.e., 15 digits), and the resulting value may be rounded to the nearest integer. One does not need to rely on this estimation scheme if one is willing to continue computation until the quadrature results from two successive levels are in agreement (to within the last few digits).

Some additional details are given in [4]. Sample implementations in Fortran-90 and C++ are available at `http://crd.lbl.gov/~dhbailey/mpdist`.

References

[1] David H. Bailey and Jonathan M. Borwein, "Effective Error Bounds in Euler-Maclaurin Based Quadrature Schemes," available at `http://crd.lbl.gov/~dhbailey/dhbpapers/em-error.pdf`.

[2] David H. Bailey and Jonathan M. Borwein, "Highly Parallel, High-Precision Numerical Integration," manuscript, 2005, available from the URL `http://crd.lbl.gov/~dhbailey/dhbpapers/quadparallel.pdf`.

[3] David H. Bailey, Jonathan M. Borwein and Richard Crandall, "Integrals of the Ising Class," manuscript in preparation.

[4] David H. Bailey, Xiaoye S. Li and Karthik Jeyabalan, "A Comparison of Three High-Precision Quadrature Schemes," *Experimental Mathematics*, vol. 14 (2005), no. 3, pg. 317–329; preprint available at `http://crd.lbl.gov/~dhbailey/dhbpapers/quadrature.pdf`.

[5] H. Takahasi and M. Mori, "Double Exponential Formulas for Numerical Integration," Publications of RIMS, Kyoto University, vol. 9 (1974), pg. 721–741.