High-Precision Numerical Integration: Progress and Challenges

D.H. Bailey* J.M. Borwein†

October 6, 2008

Abstract. One of the most fruitful advances in the field of experimental mathematics has been the development of practical methods for very high-precision numerical integration, a quest initiated by Keith Geddes and other researchers in the 1980s and 1990s. These techniques, when coupled with equally powerful integer relation detection methods, have resulted in the analytic evaluation of many integrals that previously were beyond the realm of symbolic techniques. This paper presents a survey of the current state-of-the-art in this area (including results by the present authors and others), mentions some new results, and then sketches what challenges lie ahead.

1 Introduction

Numerical evaluation of definite integrals (often termed "quadrature") has numerous applications in applied mathematics, particularly in fields such as mathematical physics and computational chemistry. Beginning in the 1980s and 1990s, researchers such as Keith Geddes and others began to explore ways to extend some of the many known techniques to the realm of high precision — tens or hundreds of digits beyond the realm of standard machine precision. A recent paper by Keith Geddes and his coauthors [19] provides a good summary of how hybrid multi-dimensional integration can be done to moderate precision within *Maple*, a subject which has been of long-term interest to Keith Geddes [20]. In more recent years related methods, one of which we will describe below, have been developed to evaluate definite integrals to an *extreme precision* of thousands of digits accuracy.

*Lawrence Berkeley National Laboratory, Berkeley, CA 94720, `dhbailey@lbl.gov`. 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-AC02-05CH11231.

†School of Mathematical and Physical Sciences, University of Newcastle, NSW, 2308, Australia and Faculty of Computer Science, Dalhousie University, Halifax, NS, B3H 2W5, Canada, `jborwein@cs.dal.ca, jonathan.borwein@newcastle.edu.au`. Supported in part by NSERC and the Canada Research Chair Programme.

These techniques have now become a staple in the emerging discipline of experimental mathematics, namely the application of high-performance computing to research questions in mathematics. In particular, high-precision numerical values of definite integrals, when combined with integer relation detection algorithms, can be used to discover previously unknown analytic evaluations (i.e., closed-form formulas) for these integrals, as well as previously unknown interrelations among classes of integrals. Large computations of this type can further be used to provide strong experimental validation of identities found in this manner or by other techniques.

We wish to emphasize that in this study we are principally *consumers* of symbolic computing tools rather than developers. However, we note that the techniques described here can be seen as an indirect means of symbolic computing — using *numerical* computation to produce *symbolic* results.

The underlying reason why very high precision results are needed in this context is that they are required if one wishes to apply *integer relation detection* methods. An integer relation detection scheme is a numerical algorithm which, given an n-long vector (x_i) of high-precision floating-point values, attempts to recover the integer coefficients (a_i), not all zero, such that

$$a_1 x_1 + a_2 x_2 + \cdots + a_n x_n \quad = \quad 0$$

(to available precision), or else determine that there are no such integers less than a certain size. The algorithm *PSLQ* operates by developing, iteration by iteration, an integer-valued matrix A which successively reduces the maximum absolute value of the entries of the vector $y = Ax$, until one of the entries of y is zero (or within an "epsilon" of zero corresponding to the level of numeric precision being used). With PSLQ or any other integer relation detection scheme, if the underlying integer relation vector of length n has entries of maximum size d digits, then the input data must be specified to at least nd-digit precision and this level of precision must also be used in the operation of the integer relation algorithm, or else the true relation will be lost in a sea of spurious numerical artifacts.

1.1 Preliminary Applications

Example 1. A Parametric Integral. In one of the first applications of this methodology, the present authors and Greg Fee of Simon Fraser University in Canada were inspired by a then recent problem in the *American Mathematical Monthly* [2]. We found, by using a version of the integration scheme described below, coupled with a high-precision PSLQ program, that if $Q(a)$ is defined by

$$Q(a) \quad := \quad \int_0^1 \frac{\arctan(\sqrt{x^2+a^2})\,dx}{\sqrt{x^2+a^2}(x^2+1)},$$

then

$$\begin{aligned}
Q(0) &= \pi \log 2/8 + G/2 \\
Q(1) &= \pi/4 - \pi\sqrt{2}/2 + 3\sqrt{2}\arctan(\sqrt{2})/2 \\
Q(\sqrt{2}) &= 5\pi^2/96.
\end{aligned}$$

Here $G = \sum_{k\geq 0}(-1)^k/(2k+1)^2 = 0.91596559417\ldots$ is *Catalan's constant*. These specific experimental results then led to the following general result, which now has been rigorously established, among several others [15, pg. 307]:

$$\int_0^\infty \frac{\arctan(\sqrt{x^2+a^2})\,dx}{\sqrt{x^2+a^2}(x^2+1)} = \frac{\pi}{2\sqrt{a^2-1}}\left[2\arctan(\sqrt{a^2-1}) - \arctan(\sqrt{a^4-1})\right].$$

■

Example 2. A Character Sum. As a second example, to assist with a study of two-variable *character Euler-sums* [18], we empirically determined that

$$\begin{aligned}
\frac{2}{\sqrt{3}}\int_0^1 \frac{\log^6(x)\arctan[x\sqrt{3}/(x-2)]}{x+1}\,dx &= \frac{1}{81648}\left[-229635L_{-3}(8)\right. \\
&+29852550L_{-3}(7)\log 3 - 1632960L_{-3}(6)\pi^2 + 27760320L_{-3}(5)\zeta(3) \\
&-275184L_{-3}(4)\pi^4 + 36288000L_{-3}(3)\zeta(5) - 30008L_{-3}(2)\pi^6 \\
&\left.-57030120L_{-3}(1)\zeta(7)\right],
\end{aligned}$$

where

$$L_{-3}(s) := \sum_{n\geq 1}\left[1/(3n-2)^s - 1/(3n-1)^s\right]$$

is based on the character modulo 3, and $\zeta(s) = \sum_{n\geq 1} 1/n^s$ is the Riemann zeta function. Based on these experimental results, general results of this type have been conjectured but few have yet been rigorously established—and may well never be unless their proof is somehow required. ■

2 High-Precision Numerical Quadrature

Several more challenging examples will be described below, but first we discuss our preferred underlying one-dimensional numerical techniques for high- and extreme-precision integration.

In our experience, we have come to rely on two schemes: *Gaussian* quadrature and *tanh-sinh* quadrature. Each has advantages in certain realms. We have found that if the function to be integrated is regular, both within the interval of integration as well as at the endpoints, and if the precision level is not too high (no more than a few hundred digits), then Gaussian quadrature is usually the most efficient scheme, in terms of achieving a desired accuracy in the lowest computer run time. In other cases, namely where the function is not regular at end points, or for precision levels above several hundred digits, tanh-sinh is usually the best choice.

2.1 Gaussian Quadrature

By way of a brief review, Gaussian quadrature approximates an integral on $[-1,1]$ as the sum $\sum_{0\le j<n} w_j f(x_j)$, where the abscissas x_j are the roots of the n-th degree Legendre polynomial $P_n(x)$ on $[-1,1]$, and the weights w_j are

$$w_j \quad := \quad \frac{-2}{(n+1)P_n'(x_j)P_{n+1}(x_j)},$$

see [3, pg. 187]. Note that the abscissas and weights are independent of $f(x)$.

In our high-precision implementations, we compute an individual abscissa by using a Newton iteration root-finding algorithm with a dynamic precision scheme. The starting value for x_j in these Newton iterations is given by $\cos[\pi(j-1/4)/(n+1/2)]$, which may be calculated using ordinary 64-bit floating-point arithmetic [22, pg. 125]. We compute the Legendre polynomial function values using an n-long iteration of the recurrence $P_0(x)=0,\ P_1(x)=1$ and

$$(k+1)P_{k+1}(x) \quad = \quad (2k+1)xP_k(x) - kP_{k-1}(x)$$

for $k \ge 2$. The derivative is computed as $P_n'(x) = n(xP_n(x)-P_{n-1}(x))/(x^2-1)$. For functions defined on intervals other than $[-1,1]$, a linear scaling is used to convert the Gaussian abscissas to the actual interval.

In our implementations, we precompute several sets of abscissa-weight pairs corresponding to values of n that roughly double with each "level." Then in a quadrature calculation we repeat Gaussian quadrature with successively higher "levels" until either two consecutive results are equal within a specified accuracy, or an estimate of the error is below the specified accuracy. In most cases, once a modest precision has been achieved, increasing the level by one (i.e., doubling n) roughly doubles the number of correct digits in the result. Additional details of an efficient, robust high-precision implementation are given in [12].

One factor that limits the applicability of Gaussian quadrature for very high precision is that the cost of computing abscissa-weight pairs using this scheme increases quadratically with n, since each Legendre polynomial evaluation requires n steps. Since the value of n required to achieve a given precision level typically increases linearly with the precision level, this means that the total run-time cost of computing the abscissas and weights increases even faster than n^2 (in fact, it increases at least as $n^3 \log n$ or n^4, depending on how the high-precision arithmetic is implemented). There is no known scheme for generating Gaussian abscissa-weight pairs that avoids this quadratic dependence on n. High-precision abscissas and weights, once computed, may be stored for future use. But for truly extreme precision calculations — i.e., several thousand digits or more — the cost of computing them even once becomes prohibitive.

2.2 Tanh-Sinh Quadrature

The other quadrature algorithm we will mention here is the tanh-sinh scheme, which was originally discovered by Takahasi and Mori [23]. This scheme, as we will see, rapidly produces very high-precision results even if the integrand function has an infinite derivative or blow-up singularity at one or both endpoints. Its other major advantage is that the cost of generating abscissas and weights increases only linearly with the number of such pairs. For these reasons, the tanh-sinh scheme is the algorithm of choice for integrating any type of function, well-behaved or not, once the precision level rises beyond several hundred digits.

The tanh-sinh scheme is based on the observation, rooted in the *Euler-Maclaurin summation* formula, that for certain bell-shaped integrands (i.e., where the function and all higher derivatives rapidly approach zero at the endpoints of the interval), a simple block-function or trapezoidal approximation to the integral is remarkably accurate [3, pg. 180]. This principle is exploited in the tanh-sinh scheme by transforming an integral of a given function $f(x)$ on a finite interval such as $[-1, 1]$ to an integral on $(-\infty, \infty)$, by using the change of variable $x = g(t)$, where $g(t) = \tanh(\pi/2 \cdot \sinh t)$. The function $g(t)$ has the property that $g(x) \to 1$ as $x \to \infty$ and $g(x) \to -1$ as $x \to -\infty$, and also that $g'(x)$ and all higher derivatives rapidly approach zero for large positive and negative arguments. Thus one can write, for $h > 0$,

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

where the abscissas $x_j = g(hj)$, the weights $w_j = g'(hj)$, and N is chosen large enough that terms beyond N (positive or negative) are smaller than the "epsilon" of the numeric precision being used. In many cases, even where $f(x)$ has an infinite derivative or an integrable singularity at one or both endpoints, the transformed integrand $f(g(t))g'(t)$ is a smooth bell-shaped function for which the Euler-Maclaurin argument applies. In these cases, the error in this approximation decreases more rapidly than any fixed power of h.

In our implementations, we typically compute sets of abscissa-weight pairs for several values of h, starting at one and then decreasing by a factor of two with each "level." The abscissa-weight pairs (and corresponding function values) at one level are merely the even-indexed pairs (and corresponding function values) for the next higher level, so this fact can be used to save run time. We terminate the quadrature calculation when two consecutive levels have produced the same quadrature results to a specified accuracy, or when an estimate of the error is below the specified accuracy. Full details of an efficient, robust high-precision implementation are given in [12].

Like Gaussian quadrature, tanh-sinh quadrature often achieves "quadratic" or "exponential" convergence — in most cases, once a modest accuracy has been achieved, increasing the "level" by one (i.e., reducing h by half and roughly doubling N), approximately doubles the number of correct digits in the result. A proof of this fact, given certain regularity conditions, is discussed in [24].

2.3 Error Estimation

As suggested above, we often rely on estimates of the error in a quadrature calculation, terminating the process when the estimated error is less than a prespecified accuracy. In several cases we have desired a more rigorous bound on the error, so that we can produce a "certificate" that the computed result is correct to within a given accuracy. In other cases, it is well worth avoiding the cost of the additional final step by use of an ad hoc estimate, such as the one described in [12].

There are several well-known rigorous error estimates for Gaussian quadrature [3, pg. 279]. Recently we established some similarly rigorous error estimates for tanh-sinh quadrature and other Euler-Maclaurin-based quadrature schemes, estimates that are both highly accurate and can be calculated in a similar manner to the quadrature result itself [4]. For example,

$$E_2(h,m) \quad := \quad h(-1)^{m-1}\left(\frac{h}{2\pi}\right)^{2m} \sum_{j=a/h}^{b/h} D^{2m}[g'(t)f(g(t))](jh) \tag{1}$$

yields extremely accurate estimates of the error, even in the simplest case $m = 1$. What's more, one can derive the bound

$$\begin{aligned} &|E(h,m) - E_2(h,m)| \quad \leq \\ &2\left[\zeta(2m) + (-1)^m\zeta(2m+2)\right]\left(\frac{h}{2\pi}\right)^{2m} h\sqrt{\int_a^b \left|D^{2m}[g'(t)f(g(t))]\right|^2 dt}, \end{aligned} \tag{2}$$

where $E(h,m)$ is the true error. These error bounds can be used to obtain rigorous certificates on the values of quadrature results.

Example 3. A Hyperbolic Volume. By using these methods we were able to establish rigorously that the experimentally observed identity

$$\begin{aligned} \frac{24}{7\sqrt{7}}\int_{\pi/3}^{\pi/2} \log\left|\frac{\tan t + \sqrt{7}}{\tan t - \sqrt{7}}\right| dt \quad &= \quad L_{-7}(2) \\ &= \sum_{n=0}^{\infty}\left[\frac{1}{(7n+1)^2} + \frac{1}{(7n+2)^2}\right. \\ &\left. -\frac{1}{(7n+3)^2} + \frac{1}{(7n+4)^2} - \frac{1}{(7n+5)^2} - \frac{1}{(7n+6)^2}\right] \end{aligned} \tag{3}$$

holds to within 3.82×10^{-49}. We subsequently learned an equivalent form of this identity was proved in 1986 [25]. ■

Other examples of such certificates are given in [4]. Obtaining these certificate results is typically rather expensive, but by applying highly parallel implementation techniques these results can be obtained in reasonable run time.

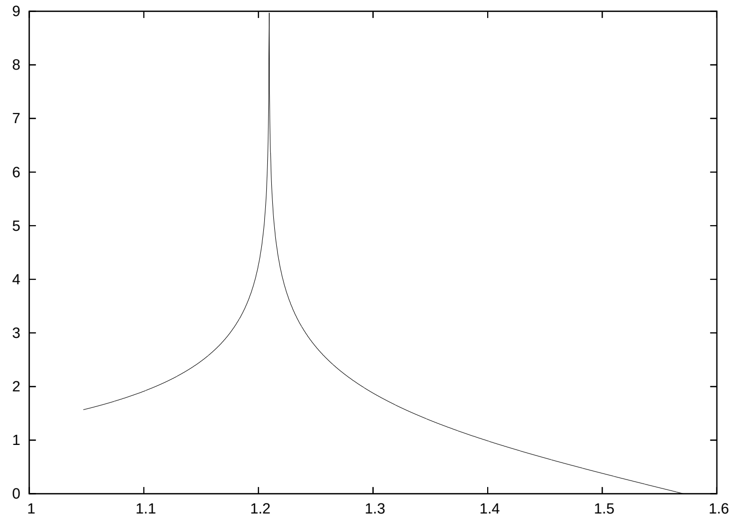

Figure 1: **Integrand function in (3) with singularity**

2.4 Highly Parallel Quadrature

Both Gaussian quadrature and tanh-sinh quadrature are well-suited to highly parallel implementations, since each of the individual abscissa-weight calculations can be performed independently, as can each of the terms of the quadrature summation. One difficulty, however, is that if one is proceeding from level to level until a certain accuracy condition is met, then a serious load imbalance may occur if a cyclic distribution of abscissa-weight pairs is adopted. This can be solved by distributing these pairs in a more intelligent fashion [5].

Example 4. The Hyperbolic Volume. In the previous section, we introduced the conjectured identity (3). Only quite recently did we learn that an equivalent form had been proven in 1986 by Don Zagier [25]. The integral arises from quantum field theory, in analysis of the volumes of ideal tetrahedra in hyperbolic space [16]. Note that the integrand function has a nasty singularity at $t = \arctan(\sqrt{7})$ (see Figure 1).

Because there are roughly a thousand related and unresolved conjectures—of which (3) was the simplest—evaluating hyperbolic volumes as arithmetic values [16], we decided to calculate the integral

$$\frac{24}{7\sqrt{7}} \int_{\pi/3}^{\pi/2} \log \left| \frac{\tan t + \sqrt{7}}{\tan t - \sqrt{7}} \right| dt \quad = \quad 1.15192547054449104710169239732054996\ldots$$

to 20,000-digit accuracy (which approaches the limits of presently feasible computation) and compare with a 20,000-digit evaluation of the six-term infinite series on the right-hand side of (3). This integral was evaluated by splitting it into two integrals, the first from $\pi/3$ to $\arctan(\sqrt{7})$, and the second from $\arctan(\sqrt{7})$ to $\pi/2$, and then applying the 1-D tanh-sinh scheme to each part, performed in parallel on a highly parallel computer system. This test was successful—the numerical value of the integral on the left-hand side of (3) agrees with the numerical value of the six-term infinite series on the right-hand side to at least 19,995 digits. The infinite series was evaluated in approximately five hours on a personal computer using *Mathematica*.

The computation of the integral was performed on the Apple-based parallel computer system at Virginia Tech. For this calculation, as well as for a number of others we describe below, we utilized the ARPREC arbitrary precision software package [11]. Parallel execution was controlled using the Message Passing Interface (MPI) software, a widely available system for parallel scientific computation [21]. The run required 45 minutes on 1024 CPUs of the Virginia Tech system, and ran at 690 Gflop/s (i.e., 690 billion 64-bit floating-point operations per second). It achieved an almost perfect parallel speedup of 993 (a perfect speedup would be 1024). Additional details of parallel quadrature techniques, for both one-dimensional and multi-dimensional integrals, are given in [5]. ■

Example 5. A Bessel Function integral. In a 2008 study, the present authors together with mathematical physicists David Broadhurst and Larry Glasser, examined various classes of integrals involving Bessel functions [6]. In this process we examined the constant $s_{6,1}$ defined by

$$s_{6,1} \quad = \quad \int_0^\infty t I_0(t) K_0^5(t)\, dt,$$

where $I_0(t)$ and $K_0(t)$ are the *modified Bessel functions* of the first and second kinds, respectively. This constant can be evaluated to high-precision by applying quadrature to this definition, or, even faster, by using a rapidly converging summation formula as described in [6]. Subsequently we discovered, by numerical experimentation, that

$$12 s_{6,1} \quad = \quad \pi^4 \int_0^\infty e^{-4t} I_0^4(t)\, dt. \tag{4}$$

Note that the integrand in (4) is singular at the origin (see Figure 2).

In order to test this conjecture, the present authors computed the right-hand side integral (4) to 14,285 digits, using a modified version of the parallel tanh-sinh program described above. This calculation required a run of 99 minutes on 1024 CPUs of the "Franklin" system (a Cray XT4 model, based on dual-core Opteron CPUs) in the National Energy Research Scientific Computing Center (NERSC) at the Lawrence Berkeley National Laboratory. These digits exactly matched the first 14,285 digits of $s_{6,1}$ computed by Broadhurst using the infinite series formula. Broadhurst now reports that he has a proof for (4). ■

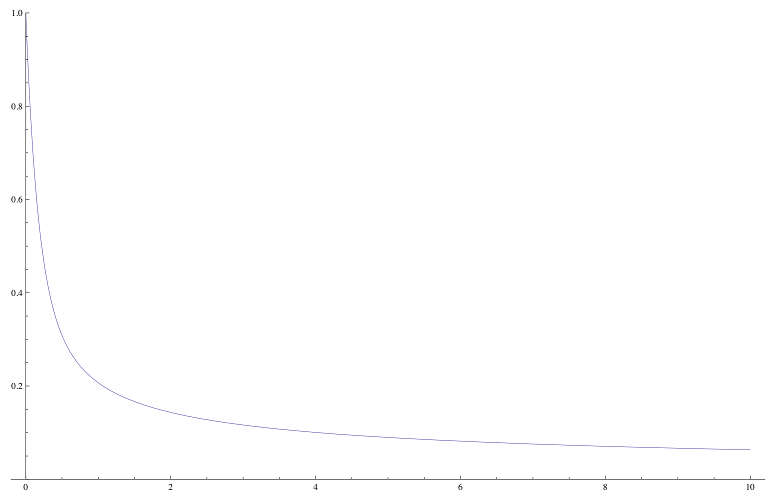

Figure 2: **Integrand function in (4) with singularity**

2.5 Multi-Dimensional Integration

The above examples are ordinary one-dimensional integrals, but two-dimensional, three-dimensional and higher-dimensional integrals are of at least as much interest. At present, the best available scheme for such computation is simply to perform either Gaussian quadrature or tanh-sinh quadrature in each dimension. Such computations are however vastly more expensive than in one dimension. For instance, if in one dimension a certain class of definite integral requires, say, 1000 function evaluations to achieve a certain desired accuracy, then in two dimensions one can expect to perform roughly 1,000,000 function evaluations for similar types of integrals, and 1,000,000,000 evaluations in three dimensions.

Fortunately, modern highly parallel computing technology makes it possible to consider the evaluation of some multiple integrals that heretofore would not have been feasible. Some of the techniques used in highly parallel, multi-dimensional integration are discussed in [5]. They typically require significant symbolic work prior to numerical computation.

3 Application to Ising Integrals

In a recent study, the present authors together with Richard Crandall examined the following classes of integrals, which arise in the Ising theory of mathematical physics [7]:

$$\begin{aligned}
C_n &:= \frac{4}{n!}\int_0^\infty \cdots \int_0^\infty \frac{1}{\left(\sum_{j=1}^n (u_j + 1/u_j)\right)^2}\frac{du_1}{u_1}\cdots\frac{du_n}{u_n} \\
D_n &:= \frac{4}{n!}\int_0^\infty \cdots \int_0^\infty \frac{\prod_{i<j}\left(\frac{u_i-u_j}{u_i+u_j}\right)^2}{\left(\sum_{j=1}^n (u_j + 1/u_j)\right)^2}\frac{du_1}{u_1}\cdots\frac{du_n}{u_n} \\
E_n &= 2\int_0^1 \cdots \int_0^1 \left(\prod_{1\le j<k\le n}\frac{u_k-u_j}{u_k+u_j}\right)^2 dt_2\,dt_3\cdots dt_n,
\end{aligned}$$

where (in the last line) $u_k = \prod_{i=1}^k t_i$.[1]

Needless to say, evaluating these n-dimensional integrals to high precision presents a daunting challenge.

3.1 The case of C_n

Example 6. The Limit of C_n. Fortunately, in the first case, we were able to show that the C_n integrals can be written as one-dimensional integrals:

$$C_n = \frac{2^n}{n!}\int_0^\infty pK_0^n(p)\ dp,$$

where K_0 is the *modified Bessel function* [1]. We were able to identify the first few instances of $C(n)$ in terms of well-known constants. For instance,

$$\begin{aligned}
C_3 &= \mathrm{L}_{-3}(2) = \sum_{n\ge 0}\left(\frac{1}{(3n+1)^2}-\frac{1}{(3n+2)^2}\right) \\
C_4 &= 14\zeta(3).
\end{aligned}$$

When we computed C_n for fairly large n, e.g.

$$C_{1024} = 0.6304735033743867961220401927108789043545870787127323\ldots$$

we found that these values rather quickly approached a limit. By using the new edition of the *Inverse Symbolic Calculator*, available at

`http://ddrive.cs.dal.ca/~isc`

this can be numerically identified as

$$\lim_{n\to\infty} C_n = 2e^{-2\gamma}.$$

We later we able to prove this fact—indeed this is merely the first term of an asymptotic expansion [7]. ■

[1] There are corresponding $(n-1)$-dimensional representations for C_n and D_n.

3.2 The case of D_n and E_n

The more fundamental integrals D_n and E_n proved to be much more difficult to evaluate. They are *not* reducible to one-dimensional integrals (as far as we can tell), but with certain symmetry transformations and symbolic integration we were able to reduce the dimension in each case by one or two, so that we were able to produce the following evaluations, all of which save the last we subsequently were able to prove:

$$
\begin{aligned}
D_2 &= 1/3 \\
D_3 &= 8 + 4\pi^2/3 - 27\,\mathrm{L}_{-3}(2) \\
D_4 &= 4\pi^2/9 - 1/6 - 7\zeta(3)/2 \\
E_2 &= 6 - 8\log 2 \\
E_3 &= 10 - 2\pi^2 - 8\log 2 + 32\log^2 2 \\
E_4 &= 22 - 82\zeta(3) - 24\log 2 + 176\log^2 2 - 256(\log^3 2)/3 \\
&\quad +16\pi^2\log 2 - 22\pi^2/3 \\
E_5 &\stackrel{?}{=} 42 - 1984\,\mathrm{Li}_4(1/2) + 189\pi^4/10 - 74\zeta(3) - 1272\zeta(3)\log 2 \\
&\quad +40\pi^2\log^2 2 - 62\pi^2/3 + 40(\pi^2\log 2)/3 + 88\log^4 2 \\
&\quad +464\log^2 2 - 40\log 2.
\end{aligned}
$$

In the case of D_n these were confirmations of known results.

Example 7. The Integral E_5. The result for E_5 required considerable effort, both computational and analytical. We did find a transformation that reduced this to a 3-D integral, but the resulting 3-D integral is extremely complicated (see Table 1). Just converting this expression (originally produced as a *Mathematica* expression) to a working computer program required considerable ingenuity. The numerical evaluation of this integral to 240 digits required four hours on 64 CPUs of the Virginia Tech Apple system. Applying PSLQ to the resulting numerical value (together with the numerical values of a set of conjectured terms), yielded the experimental evaluation shown above. ■

3.3 A Negative Result

Example 8. The Integrals C_5, and D_5. The computation of D_5 is even more demanding than E_5. Nonetheless, 18 hours on 256 CPUs of the Apple system at Virginia Tech produced 500 good digits. Alas, we still have not been successful in identifying either C_5 or D_5 from their numerical values (or by any other means). However, we have shown, via PSLQ computations, that neither C_5 nor D_5 satisfies an integer linear relation involving the following set of constants, where the vector of integer coefficients in the linear relation has Euclidean norm less than $4 \cdot 10^{12}$:

$1,\ \pi,\ \log 2,\ \pi^2,\ \pi\log 2,\ \log^2 2,\ \mathrm{L}_{-3}(2),\ \pi^3,\ \pi^2\log 2,\ \pi\log^2 2,\ \log^3 2,$
$\zeta(3),\ \pi\,\mathrm{L}_{-3}(2),\ \log 2\cdot\mathrm{L}_{-3}(2),\ \pi^4\ \pi^3\log 2,\ \pi^2\log^2 2,\ \pi\log^3 2,\ G,\ G\pi^2,$

$$
\begin{aligned}
E_5 \quad = \quad & \int_0^1 \int_0^1 \int_0^1 \left[2(1-x)^2(1-y)^2(1-xy)^2(1-z)^2(1-yz)^2(1-xyz)^2\right. \\
& \left(-\left[4(x+1)(xy+1)\log(2)\left(y^5z^3x^7 - y^4z^2(4(y+1)z+3)x^6 - y^3z\left(\left(y^2+1\right)z^2+4(y+\right.\right.\right.\right. \\
& 1)z+5\big)x^5 + y^2\left(4y(y+1)z^3+3\left(y^2+1\right)z^2+4(y+1)z-1\right)x^4 + y\left(z\left(z^2+4z\right.\right. \\
& +5\big)y^2+4\left(z^2+1\right)y+5z+4\big)x^3 + \left(\left(-3z^2-4z+1\right)y^2-4zy+1\right)x^2 - (y(5z+4) \\
& +4)x-1)\big] / \left[(x-1)^3(xy-1)^3(xyz-1)^3\right] + \left[3(y-1)^2y^4(z-1)^2z^2(yz\right. \\
& -1)^2x^6 + 2y^3z\left(3(z-1)^2z^3y^5 + z^2\left(5z^3+3z^2+3z+5\right)y^4 + (z-1)^2z\right. \\
& \left(5z^2+16z+5\right)y^3 + \left(3z^5+3z^4-22z^3-22z^2+3z+3\right)y^2 + 3\left(-2z^4+z^3+2\right. \\
& z^2+z-2\big)y + 3z^3+5z^2+5z+3\big)x^5 + y^2\left(7(z-1)^2z^4y^6 - 2z^3\left(z^3+15z^2\right.\right. \\
& +15z+1\big)y^5 + 2z^2\left(-21z^4+6z^3+14z^2+6z-21\right)y^4 - 2z\left(z^5-6z^4-27z^3\right. \\
& -27z^2-6z+1\big)y^3 + \left(7z^6-30z^5+28z^4+54z^3+28z^2-30z+7\right)y^2 - 2\left(7z^5\right. \\
& +15z^4-6z^3-6z^2+15z+7\big)y + 7z^4-2z^3-42z^2-2z+7\big)x^4 - 2y\left(z^3\left(z^3\right.\right. \\
& -9z^2-9z+1\big)y^6 + z^2\left(7z^4-14z^3-18z^2-14z+7\right)y^5 + z\left(7z^5+14z^4+3\right. \\
& z^3+3z^2+14z+7\big)y^4 + \left(z^6-14z^5+3z^4+84z^3+3z^2-14z+1\right)y^3 - 3\left(3z^5\right. \\
& +6z^4-z^3-z^2+6z+3\big)y^2 - \left(9z^4+14z^3-14z^2+14z+9\right)y + z^3+7z^2+7z \\
& +1\big)x^3 + \left(z^2\left(11z^4+6z^3-66z^2+6z+11\right)y^6 + 2z\left(5z^5+13z^4-2z^3-2z^2\right.\right. \\
& +13z+5\big)y^5 + \left(11z^6+26z^5+44z^4-66z^3+44z^2+26z+11\right)y^4 + \left(6z^5-4\right. \\
& z^4-66z^3-66z^2-4z+6\big)y^3 - 2\left(33z^4+2z^3-22z^2+2z+33\right)y^2 + \left(6z^3+26\right. \\
& z^2+26z+6\big)y + 11z^2+10z+11\big)x^2 - 2\left(z^2\left(5z^3+3z^2+3z+5\right)y^5 + z\left(22z^4\right.\right. \\
& +5z^3-22z^2+5z+22\big)y^4 + \left(5z^5+5z^4-26z^3-26z^2+5z+5\right)y^3 + \left(3z^4-\right. \\
& 22z^3-26z^2-22z+3\big)y^2 + \left(3z^3+5z^2+5z+3\right)y + 5z^2+22z+5\big)x + 15z^2+2z \\
& +2y(z-1)^2(z+1) + 2y^3(z-1)^2z(z+1) + y^4z^2\left(15z^2+2z+15\right) + y^2\left(15z^4\right. \\
& -2z^3-90z^2-2z+15\big)+15\big] / \left[(x-1)^2(y-1)^2(xy-1)^2(z-1)^2(yz-1)^2\right. \\
& (xyz-1)^2\big] - \left[4(x+1)(y+1)(yz+1)\left(-z^2y^4+4z(z+1)y^3+\left(z^2+1\right)y^2\right.\right. \\
& -4(z+1)y+4x\left(y^2-1\right)\left(y^2z^2-1\right)+x^2\left(z^2y^4-4z(z+1)y^3-\left(z^2+1\right)y^2\right. \\
& +4(z+1)y+1\big)-1\big)\log(x+1)\big] / \left[(x-1)^3x(y-1)^3(yz-1)^3\right] - [4(y+1)(xy \\
& +1)(z+1)\left(x^2\left(z^2-4z-1\right)y^4 + 4x(x+1)\left(z^2-1\right)y^3 - \left(x^2+1\right)\left(z^2-4z-1\right)\right. \\
& y^2 - 4(x+1)\left(z^2-1\right)y + z^2-4z-1\big)\log(xy+1)\big] / \left[x(y-1)^3y(xy-1)^3(z-\right. \\
& 1)^3\big] - \left[4(z+1)(yz+1)\left(x^3y^5z^7 + x^2y^4(4x(y+1)+5)z^6 - xy^3\left(\left(y^2+\right.\right.\right.\right. \\
& 1\big)x^2-4(y+1)x-3\big)z^5 - y^2\left(4y(y+1)x^3+5\left(y^2+1\right)x^2+4(y+1)x+1\right)z^4 + \\
& y\left(y^2x^3-4y(y+1)x^2-3\left(y^2+1\right)x-4(y+1)\right)z^3 + \left(5x^2y^2+y^2+4x(y+1)\right. \\
& y+1\big)z^2 + ((3x+4)y+4)z-1\big)\log(xyz+1)\big] / \left[xy(z-1)^3z(yz-1)^3(xyz-1)^3\right]\big)\big] \\
& / \left[(x+1)^2(y+1)^2(xy+1)^2(z+1)^2(yz+1)^2(xyz+1)^2\right] \; dx\, dy\, dz
\end{aligned}
$$

Table 1: **The E_5 integral**

Li$_4(1/2)$, $\sqrt{3}\,\mathrm{L}_{-3}(2)$, $\log^4 2$, $\pi\zeta(3)$, $\log 2\cdot\zeta(3)$, $\pi^2\mathrm{L}_{-3}(2)$, $\pi^2\mathrm{L}_{-3}(2)$, $\pi\log 2{\cdot}\mathrm{L}_{-3}(2)$, $\log^2 2{\cdot}\mathrm{L}_{-3}(2)$, $\mathrm{L}^2_{-3}(2)$, $\mathrm{Im}[\mathrm{Li}_4(e^{2\pi i/5})]$, $\mathrm{Im}[\mathrm{Li}_4(e^{4\pi i/5})]$, $\mathrm{Im}[\mathrm{Li}_4(i)]$, $\mathrm{Im}[\mathrm{Li}_4(e^{2\pi i/3})]$.

Here G and L_{-3} are as above and $\mathrm{Li}_n(x) := \sum_{k\geq 1} x^k/k^n$ is the *polylogarithm* function. Some constants that may appear to be "missing" from this list are actually linearly redundant with this set, and thus were not included in the PSLQ search. These include

$\mathrm{Re}[\mathrm{Li}_3(i)]$, $\mathrm{Im}[\mathrm{Li}_3(i)]$, $\mathrm{Re}[\mathrm{Li}_3(e^{2\pi i/3})]$, $\mathrm{Im}[\mathrm{Li}_3(e^{2\pi i/3})]$, $\mathrm{Re}[\mathrm{Li}_4(i)]$, $\mathrm{Re}[\mathrm{Li}_4(e^{2\pi i/3})]$, $\mathrm{Re}[\mathrm{Li}_4(e^{2\pi i/5})]$, $\mathrm{Re}[\mathrm{Li}_4(e^{4\pi i/5})]$, $\mathrm{Re}[\mathrm{Li}_4(e^{2\pi i/6})]$ and $\mathrm{Im}[\mathrm{Li}_4(e^{2\pi i/6})]$.

Despite this failure, we view these computations as successful, since the numerical values of D_5 and C_5 (along with numerous other related constants) are available at [8] to any researcher who may have a better idea of where to hunt for a closed form. ■

3.4 Closed Forms for $c_{n,k}$

In a follow-on study [9], the present authors and Richard Crandall examined the following generalization of the C_n integrals above:

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

Here we made the initially surprising discovery—now proven in [17]—that there are linear relations in each of the rows of this array (considered as a doubly-infinite rectangular matrix). For instance,

$$\begin{aligned}
0 &= C_{3,0} - 84C_{3,2} + 216C_{3,4} \\
0 &= 2C_{3,1} - 69C_{3,3} + 135C_{3,5} \\
0 &= C_{3,2} - 24C_{3,4} + 40C_{3,6} \\
0 &= 32C_{3,3} - 630C_{3,5} + 945C_{3,7} \\
0 &= 125C_{3,4} - 2172C_{3,6} + 3024C_{3,8}.
\end{aligned}$$

In yet a more recent study [6], we were able to analytically recognize many of these $C_{n,k}$ integrals—because, remarkably, these same integrals appear naturally in quantum field theory (for odd k)!

Example 10. Four Hypergeometric Forms. In particular, we discovered, and then proved with considerable effort that, with $c_{n,k}$ normalized by $C_{n,k} =$

$2^n\, c_{n,k}/(n!\, k!)$, we have

$$\begin{aligned}
c_{3,0} &= \frac{3\Gamma^6(1/3)}{32\pi 2^{2/3}} = \frac{\sqrt{3}\pi^3}{8}\,{}_3F_2\left(\begin{array}{c} 1/2, 1/2, 1/2 \\ 1, 1 \end{array} \middle| \frac{1}{4} \right) \\
c_{3,2} &= \frac{\sqrt{3}\pi^3}{288}\,{}_3F_2\left(\begin{array}{c} 1/2, 1/2, 1/2 \\ 2, 2 \end{array} \middle| \frac{1}{4} \right) \\
c_{4,0} &= \frac{\pi^4}{4} \sum_{n=0}^{\infty} \frac{\binom{2n}{n}^4}{4^{4n}} = \frac{\pi^4}{4}\,{}_4F_3\left(\begin{array}{c} 1/2, 1/2, 1/2, 1/2 \\ 1, 1, 1 \end{array} \middle| 1 \right) \\
c_{4,2} &= \frac{\pi^4}{64} \left[4\,{}_4F_3\left(\begin{array}{c} 1/2, 1/2, 1/2, 1/2 \\ 1, 1, 1 \end{array} \middle| 1 \right) \right. \\
&\quad \left. -3\,{}_4F_3\left(\begin{array}{c} 1/2, 1/2, 1/2, 1/2 \\ 2, 1, 1 \end{array} \middle| 1 \right) \right] - \frac{3\pi^2}{16},
\end{aligned}$$

where ${}_pF_q$ denotes the *generalized hypergeometric* function [1]. ■

4 Application to Elliptic Integral Evaluations

The work in [6] also required computation of some very tricky two and three-dimensional integrals arising from *sunrise diagrams* in quantum field theory.

Example 10. Two Elliptic Integral Integrals. For example, with $\mathbf{K}$ denoting the *complete elliptic integral* of the first kind, we had conjectured

$$c_{5,0} = \frac{\pi}{2} \int_{-\pi/2}^{\pi/2} \int_{-\pi/2}^{\pi/2} \frac{\mathbf{K}(\sin\theta)\,\mathbf{K}(\sin\phi)}{\sqrt{\cos^2\theta\cos^2\phi + 4\sin^2(\theta+\phi)}}\, \mathrm{d}\theta\, \mathrm{d}\phi\,. \tag{5}$$

Note that this function has singularities on all four sides of the domain of integration (see Figure 3).

Ultimately, we were able to compute $c_{5,0}$ to 120-digit accuracy, using 240-digit working precision. This run required a parallel computation (using the MPI parallel programming library) of 43.2 minutes on 512 CPUs (1024 cores) of the "Franklin" system at LBNL. Likewise we could confirm

$$c_{6,0} = \frac{\pi}{2} \int_{-\pi/2}^{\pi/2} \int_{-\pi/2}^{\pi/2} \frac{\mathbf{K}(\sin\theta)\,\mathbf{K}(\sin\phi)\,\mathbf{K}\left(\frac{\sin(\theta+\phi)}{\sqrt{\cos^2\theta\cos^2\phi+\sin^2(\theta+\phi)}}\right)}{\sqrt{\cos^2\theta\cos^2\phi + \sin^2(\theta+\phi)}}\, \mathrm{d}\theta\, \mathrm{d}\phi\ .$$

■

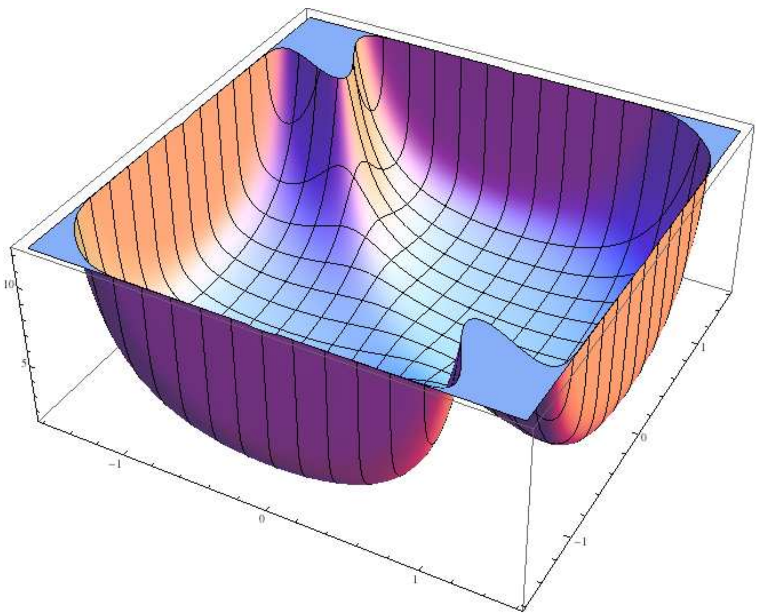

Figure 3: **Plot of $c_{5,0}$ integrand function in (5)**

Example 11. A Bessel Moment Conjecture. This same work also uncovered the following striking *conjecture*: for each integer pair (n, k) with $n \geq 2k \geq 2$ we have

$$\sum_{m=0}^{\lfloor n/2 \rfloor} (-1)^m \binom{n}{2m} \int_0^\infty t^{n-2k} [\pi I_0(t)]^{n-2m} [K_0(t)]^{n+2m} \, \mathrm{d}t \quad \stackrel{?}{=} \quad 0$$

where I_0 and K_0 are again Bessel functions. ■

5 Application to Heisenberg Spin Integrals

In another recent application of these methods, we investigated the following integrals ("spin integrals"), which arise, like the Ising integrals, from studies in mathematical physics [13, 14]:

$$\begin{aligned} P(n) \quad &:= \quad \frac{\pi^{n(n+1)/2}}{(2\pi i)^n} \cdot \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} U(x_1 - i/2, x_2 - i/2, \cdots, x_n - i/2) \\ &\quad \times T(x_1 - i/2, x_2 - i/2, \cdots, x_n - i/2) \, dx_1 dx_2 \cdots dx_n \end{aligned}$$

n	Digits	Processors	Run Time
2	120	1	10 sec.
3	120	8	55 min.
4	60	64	27 min.
5	30	256	39 min.
6	6	256	59 hrs.

Table 2: **Computation times for** $P(n)$

where

$$\begin{aligned} U(x_1 - i/2, x_2 - i/2, \cdots, x_n - i/2) &= \frac{\prod_{1\le k<j\le n} \sinh[\pi(x_j - x_k)]}{\prod_{1\le j\le n} i^n \cosh^n(\pi x_j)} \\ T(x_1 - i/2, x_2 - i/2, \cdots, x_n - i/2) &= \frac{\prod_{1\le j\le n} (x_j - i/2)^{j-1}(x_j + i/2)^{n-j}}{\prod_{1\le k<j\le n}(x_j - x_k - i)}. \end{aligned}$$

Note that these integrals involve some complex-arithmetic calculations, even though the final results are real.

Example 12. Spin Values. So far we have been able to numerically confirm the following results:

$$\begin{aligned} P(1) &= \frac{1}{2}, \quad P(2) = \frac{1}{3} - \frac{1}{3}\log 2, \quad P(3) = \frac{1}{4} - \log 2 + \frac{3}{8}\zeta(3) \\ P(4) &= \frac{1}{5} - 2\log 2 + \frac{173}{60}\zeta(3) - \frac{11}{6}\zeta(3)\log 2 - \frac{51}{80}\zeta^2(3) - \frac{55}{24}\zeta(5) + \frac{85}{24}\zeta(5)\log 2 \\ P(5) &= \frac{1}{6} - \frac{10}{3}\log 2 + \frac{281}{24}\zeta(3) - \frac{45}{2}\zeta(3)\log 2 - \frac{489}{16}\zeta^2(3) - \frac{6775}{192}\zeta(5) \\ &\quad + \frac{1225}{6}\zeta(5)\log 2 - \frac{425}{64}\zeta(3)\zeta(5) - \frac{12125}{256}\zeta^2(5) + \frac{6223}{256}\zeta(7) \\ &\quad - \frac{11515}{64}\zeta(7)\log 2 + \frac{42777}{512}\zeta(3)\zeta(7), \end{aligned}$$

as well as a significantly more complicated expression for $P(6)$. We confirmed $P(1)$ through $P(4)$ to over 60-digit precision; $P(5)$ to 30-digit precision, but $P(6)$ to only 8-digit precision. These quadrature calculations were performed as parallel jobs on the Apple G5 cluster at Virginia Tech. Were we able to compute $P(n)$ for $n < 9$ to say 100 places, we might well be able to use PSLQ to determine the precise closed forms of $P(n)$. How difficult a task this is is illustrated by the run times and processors used shown in Table 2, which underscores the rapidly escalating difficulty of these computations. ■

These huge and rapidly increasing run times, as in the Ising integral study, point to the critical need for research into fundamentally new and more efficient numerical schemes for two-dimensional and higher-dimensional integrals. It is hoped that the results in this paper will stimulate research in that direction.

References

[1] Milton Abramowitz and Irene A. Stegun, *Handbook of Mathematical Functions,* Dover, NY, 1970.

[2] Zafar Ahmed, "Definitely an Integral," *American Mathematical Monthly*, vol. 109 (2002), no. 7, pg. 670–671.

[3] K. E. Atkinson, *Elementary Numerical Analysis*, John Wiley & Sons, 1993.

[4] David H. Bailey and Jonathan M. Borwein, "Effective Error Bounds in Euler-Maclaurin-Based Quadrature Schemes," *Proc. 2006 Conf. on High-Performance Computing Systems*, IEEE Computer Society, 2006, available at `http://crd.lbl.gov/~dhbailey/dhbpapers/hpcs06.pdf`.

[5] D. H. Bailey and J. M. Borwein, "Highly Parallel, High-Precision Numerical Integration," *International Journal of Computational Science and Engineering*, to appear, 2008, available at `http://crd.lbl.gov/~dhbailey/dhbpapers/quadparallel.pdf`.

[6] D. H. Bailey, J. M. Borwein, D. Broadhurst and M. L. Glasser, "Elliptic integral evaluations of Bessel moments," *Journal of Physics A: Mathematical and General*, vol. 41 (2008), 205203 (46 pp.).

[7] D. H. Bailey, J. M. Borwein and R. E. Crandall, "Integrals of the Ising Class," *Journal of Physics A*, vol. 39 (2006), pg. 12271–12302.

[8] D. H. Bailey, J. M. Borwein and R. E. Crandall, "Ising Data," 2006, available at `http://crd.lbl.gov/~dhbailey/dhbpapers/ising-data.pdf`.

[9] David H. Bailey, David Borwein, Jonathan M. Borwein and Richard Crandall, "Hypergeometric Forms for Ising-Class Integrals," *Experimental Mathematics*, vol. 16 (2007), no. 3, pg. 257–276.

[10] David H. Bailey and David Broadhurst, "Parallel Integer Relation Detection: Techniques and Applications," *Math. of Computation*, vol. 70, no. 236 (2000), pg. 1719-1736.

[11] David H. Bailey, Yozo Hida, Xiaoye S. Li and Brandon Thompson, "ARPREC: An Arbitrary Precision Computation Package," technical report LBNL-53651, software and documentation available at `http://crdl.bl.gov/~dhbailey/mpdist`.

[12] David H. Bailey, Xiaoye S. Li and Karthik Jeyabalan, "A Comparison of Three High-Precision Quadrature Programs," *Experimental Mathematics*, vol. 14 (2005), no. 3, pg. 317–329.

[13] H. E. Boos and V. E. Korepin, "Quantum Spin Chains and Riemann Zeta Function with Odd Arguments," *Journal of Physics A*, vol. 34 (2001), pg. 5311–5316, preprint available at `http://arxiv.org/abs/hep-th/0104008`.

[14] H. E. Boos, V. E. Korepin, Y. Nishiyama and M. Shiroishi, "Quantum Correlations and Number Theory," *Journal of Physics A*, vol. 35 (2002), pg. 4443, available at `http://arxiv.org/abs/cond-mat/0202346`.

[15] Jonathan M. Borwein, David H. Bailey and Roland Girgensohn, *Experimentation in Mathematics: Computational Paths to Discovery*, A K Peters, Welesly, MA, 2004.

[16] J. Borwein and D. Broadhurst, "Determination of Rational Dirichlet-zeta Invariants of Hyperbolic Manifolds and Feynman Knots and Links," available at `http://arxiv.org/hep-th/9811173`.

[17] Jonathan Borwein and Bruno Salvy, "A Proof of a Recursion for Bessel Moments," *Experimental Mathematics*, to appear, D-drive Preprint 346, 2007, `http://locutus.cs.dal.ca:8088/archive/00000346/`.

[18] J. M. Borwein, I. J. Zucker and J. Boersma, "The Evaluation of Character Euler Double Sums," *Ramanujan Journal*, vol. 15 (2008), to appear.

[19] O. A. Carvajal, Orlando, F. W. Chapman, and K. O. Geddes, "Hybrid Symbolic-Numeric Integration in Multiple Dimensions via Tensor-Product Series," *ISSAC'05*, pg. 84–91 (electronic), ACM Press, New York, 2005.

[20] K. O. Geddes and G. J. Fee, "Hybrid Symbolic-Numeric Integration in Maple," *Proceedings of ISAAC'92*, pg. 36–41, ACM Press, New York, 1992.

[21] William Gropp, Ewing Lusk, Anthony Skjellum, *Using MPI: A Portable Parallel Programming with the Message-Passing Interface*, MIT Press, Cambridge, MA, 1996.

[22] William H. Press, Saul A. Teukolsky, William T. Vetterling, Brian P. Flannery, *Numerical Recipes 3rd Edition: The Art of Scientific Computing*, Cambridge Univ. Press, 1986.

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

[24] Lingyun Ye, *Numerical Quadrature: Theory and Computation*, MSc Thesis, Computer Science, Dalhousie University, available at `http://locutus.cs.dal.ca:8088/archive/00000328`.

[25] Don Zagier, "Hyperbolic Manifolds and Special Values of Dedekind Zeta-functions," *Inventiones Mathematicae*, vol. 83 (1986), pg. 285-301.