PROCEEDINGS OF SCIENCE

High-Precision Computation and Mathematical Physics

David H. Bailey[*] [†]
Computational Research Dept., Lawrence Berkeley National Laboratory, Berkeley, CA 94720
E-mail: dhbailey@lbl.gov

Jonathan M. Borwein[‡]
School of Mathematical And Physical Sciences, University of Newcastle, NSW 2308 Australia, and Faculty of Computer Science, Dalhousie University, Halifax, NS, B3H 2W5, Canada
Email: jonathan.borwein@newcastle.edu.au, jborwein@cs.dal.ca

At the present time, IEEE 64-bit floating-point arithmetic is sufficiently accurate for most scientific applications. However, for a rapidly growing body of important scientific computing applications, a higher level of numeric precision is required. Such calculations are facilitated by high-precision software packages that include high-level language translation modules to minimize the conversion effort. This paper presents a survey of recent applications of these techniques and provides some analysis of their numerical requirements. These applications include supernova simulations, climate modeling, planetary orbit calculations, Coulomb *n*-body atomic systems, scattering amplitudes of quarks, gluons and bosons, nonlinear oscillator theory, Ising theory, quantum field theory and experimental mathematics. We conclude that high-precision arithmetic facilities are now an indispensable component of a modern large-scale scientific computing environment.

XII Advanced Computing and Analysis Techniques in Physics Research
November 3-7, 2008
Erice, Italy

[*]Speaker.

[†]Supported in part by the Director, Office of Computational and Technology Research, Division of Mathematical, Information and Computational Sciences, U.S. Department of Energy, under contract number DE-AC02-05CH11231.

[‡]Supported in part by NSERC and the Canada Research Chair Programme.

© Copyright owned by the author(s) under the terms of the Creative Commons Attribution-NonCommercial-ShareAlike Licence.

PoS(ACAT08)014

1. Introduction

Virtually all present-day computer systems, from personal computers to the largest supercomputers, implement the IEEE 64-bit floating-point arithmetic standard, which provides 53 mantissa bits, or approximately 16 decimal digit accuracy. For most scientific applications, 64-bit arithmetic is more than sufficient, but for a rapidly expanding body of applications, it is not. In these applications, portions of the code typically involve numerically sensitive calculations, which produce results of questionable accuracy using conventional arithmetic. These inaccurate results may in turn induce other errors, such as taking the wrong path in a conditional branch.

Exacerbating these difficulties is the proliferation of very large-scale highly parallel computer systems, as as exemplified by the Top500 list (see `http://www.top500.org`). One inescapable consequence of the greatly increased scale of these calculations is that numerical anomalies which heretofore have been minor nuisances are now much more likely to have significant impact. At the same time, the majority of persons performing these computations are not experts in numerical analysis, and thus are more likely to be unaware of the potential numerical difficulties that may exist. Thus, while some may argue that numerically sensitive calculations can be remedied by using different algorithms or coding techniques, in practice it is usually easier, cheaper and more reliable to employ high-precision arithmetic to overcome them.

One concrete illustration of these difficulties is provided by the following example. Consider the very simple 1-D differential equation $y''(x) = -f(x)$ for some function $f(x)$. Discretization of this system immediately leads to the matrix

$$\begin{bmatrix} 2 & -1 & 0 & 0 & \cdots & 0 \\ -1 & 2 & -1 & 0 & \cdots & 0 \\ 0 & -1 & 2 & -1 & \cdots & 0 \\ . & . & . & . & . & . \\ 0 & \cdots & -1 & 2 & -1 & 0 \\ 0 & \cdots & 0 & -1 & 2 & -1 \\ 0 & \cdots & 0 & 0 & -1 & 2 \end{bmatrix}.$$

The condition number of this matrix (namely the quotient of the largest eigenvalue to the smallest eigenvalue) is readily seen to be approximated by

$$\kappa(n) \approx \frac{4(n+1)^2}{\pi^2},$$

where $n \times n$ is the size of the linear system above (the authors are indebted to Bastian Pentenrieder of ETH Zurich for this observation). Note that even when $n = 10^7$, which is a fairly modest size compared to some being attempted in current high-end computing, the condition number is sufficiently large that the system (depending on the nature of function $f(x)$) cannot be reliably solved using conventional IEEE 64-bit floating-point arithmetic.

2. High-Precision Software

Algorithms for performing high-precision arithmetic are fairly well known [19], and software packages implementing these schemes have been available since the early days of computing.

PoS(ACAT08)014

However, many of these packages require one to rewrite a scientific application with individual subroutine calls for each arithmetic operation. The difficulty of writing and debugging such code has deterred all but a few scientists from using such software. But in the past few years, high-precision software packages have been produced that include high-level language interfaces, making such conversions relatively painless. These packages typically utilize custom datatypes and operator overloading features, which are now available in languages such as C++ and Fortran-90, to facilitate conversion. Even more advanced high-precision computation facilities are available in the commercial products *Mathematica* and *Maple*, which incorporate arbitrary-precision arithmetic in a natural way for a wide range of functions. However, these products do not provide a means to convert existing scientific programs written in other languages.

Some examples of high-precision arithmetic software packages that are freely available on the Internet are the following, listed in alphabetical order. The ARPREC, QD and MPFUN90 packages are available from the first author's website: `http://crd.lbl.gov/~dhbailey/mpdist`.

- ARPREC. This package includes routines to perform arithmetic with an arbitrarily high level of precision, including many algebraic and transcendental functions. High-level language interfaces are available for C++ and Fortran-90, supporting real, integer and complex datatypes.

- GMP. This package includes an extensive library of routines to support high-precision integer, rational and floating-point calculations. GMP has been produced by a volunteer effort and is distributed under the GNU license by the Free Software Foundation. It is available at `http://gmplib.org`.

- MPFR. The MPFR library is a C library for multiple-precision floating-point computations with exact rounding, and is based on the GMP multiple-precision library. Additional information is available at `http://www.mpfr.org`.

- MPFR++. This is a high-level C++ interface to MPFR. Additional information is available at `http://perso.ens-lyon.fr/nathalie.revol/software.html`. A similar package is GMPFRXX, available at `http://math.berkeley.edu/~wilken/code/gmpfrxx`.

- MPFUN90. This is equivalent to ARPREC in user-level functionality, but is written entirely in Fortran-90 and provides a Fortran-90 language interface.

- QD. This package includes routines to perform "double-double" (approx. 31 digits) and "quad-double" (approx. 62 digits) arithmetic. High-level language interfaces are available for C++ and Fortran-90, supporting real, integer and complex datatypes. The QD package is much faster than using arbitrary precision software when 31 or 62 digits is sufficient.

Using high-precision software increases computer run times, compared with using conventional 64-bit arithmetic. For example, computations using double-double precision arithmetic typically run five times slower than with 64-bit arithmetic. This figure rises to 25 times for the quad-double arithmetic, to more than 50 times for 100-digit arithmetic, and to more than 1000 times for 1000-digit arithmetic.

PoS(ACAT08)014

3. Applications of High-Precision Arithmetic

Here we briefly mention a few of the growing list of scientific computations that require high-precision arithmetic, and provide some analysis of their numerical requirements.

3.1 Supernova Simulations

Recently Edward Baron, Peter Hauschildt, and Peter Nugent used the QD package, which provides double-double (128-bit or 31-digit) and quad-double (256-bit or 62-digit) datatypes, to solve for the non-local thermodynamic equilibrium populations of iron and other atoms in the atmospheres of supernovae and other astrophysical objects [15, 24]. Iron for example may exist as Fe II in the outer parts of the atmosphere, but in the inner parts Fe IV or Fe V could be dominant. Introducing artificial cutoffs leads to numerical glitches, so it is necessary to solve for all of these populations simultaneously. Since the relative population of any state from the dominant stage is proportional to the exponential of the ionization energy, the dynamic range of these numerical values can be very large.

In order to handle this potentially very large dynamic range, yet at the same time perform the computation in reasonable time, Baron, Hauschildt and Nugent employ an automatic scheme to determine whether to use 64-bit, 128-bit or 256-bit arithmetic in both constructing the matrix elements and in solving the linear system.

3.2 Climate Modeling

It is well-known that climate simulations are fundamentally chaotic—if microscopic changes are made to the present state, within a certain period of simulated time the future state is completely different. Indeed, ensembles of these calculations are required to obtain statistical confidence in global climate trends produced from such calculations. As a result, computational scientists involved in climate modeling applications have resigned themselves that their codes quickly diverge from any "baseline" calculation, even if they only change the number of processors used to run the code. For this reason, it is not only difficult for researchers to compare results, but it is often problematic even to determine whether they have correctly deployed their code on a given system.

Recently Helen He and Chris Ding investigated this non-reproducibility phenomenon in a widely-used climate modeling code. They found that almost all of the numerical variation occurred in one inner product loop in the atmospheric data assimilation step, and in a similar operation in a large conjugate gradient calculation. He and Ding found that a straightforward solution was to employ double-double arithmetic for these loops. This single change dramatically reduced the numerical variability of the entire application, permitting computer runs to be compared for much longer run times than before [25].

3.3 Planetary Orbit Calculations

One central question of planetary theory is whether the solar system is stable over cosmological time frames (billions of years). Planetary orbits well known to exhibit chaotic behavior. Indeed, as Isaac Newton once noted, "The orbit of any one planet depends on the combined motions of all the planets, not to mention the actions of all these on each other. To consider simultaneously all

PoS(ACAT08)014

these causes of motion and to define these motions by exact laws allowing of convenient calculation exceeds, unless I am mistaken, the forces of the entire human intellect." [22, pg. 121].

Scientists have studied this question by performing very long-term simulations of planetary motions. These simulations typically do fairly well for long periods, but then fail at certain key junctures, such as when two planets pass fairly close to each other. Researchers have found that double-double or quad-double arithmetic is required to avoid severe numerical inaccuracies, even if other techniques are employed to reduce numerical error [26].

3.4 Coulomb *n*-Body Atomic System Simulations

Numerous computations have been performed recently using high-precision arithmetic to study atomic-level Coulomb systems. For example, Alexei Frolov of Queen's University in Ontario, Canada has used high-precision software to solve the generalized eigenvalue problem $(\hat{H} - E\hat{S})C = 0$, where the matrices $\hat{H}$ and $\hat{S}$ are large (typically $5,000 \times 5,000$ in size) and very nearly degenerate. Until recently, progress in this arena was severely hampered by the numerical difficulties induced by these nearly degenerate matrices.

Frolov has done his calculations using the MPFUN90 package, with a numeric precision level exceeding 100 digits. Frolov notes that in this way "we can consider and solve the bound state few-body problems which have been beyond our imagination even four years ago." He has also used MPFUN90 to compute the matrix elements of the Hamiltonian matrix $\hat{H}$ and the overlap matrix $\hat{S}$ in four- and five-body atomic problems. As of this date, Frolov has written a total of 21 papers based on high-precision computations. Two illustrative examples are [13] and [23].

3.5 Studies of the Fine Structure Constant of Physics

In the past few years, significant progress has been achieved in using high-precision arithmetic to obtain highly accurate solutions to the Schrodinger equation for the lithium atom. In particular, the nonrelativistic ground state energy has been calculated to an accuracy of a few parts in a trillion, a factor of 1500 improvement over the best previous results. With these highly accurate wavefunctions, Zong-Chao Yan and others have been able to test the relativistic and QED effects at the 50 parts per million (ppm) level and also at the one ppm level [30]. Along this line, a number of properties of lithium and lithium-like ions have also been calculated, including the oscillator strengths for certain resonant transitions, isotope shifts in some states, dispersion coefficients and Casimir-Polder effects between two lithium atoms.

Theoretical calculations of the fine structure splittings in helium atoms have now advanced to the stage that highly accurate experiments are now planned. When some additional computations are completed, a unique atomic physics value of the fine structure constant may be obtained to an accuracy of 16 parts per billion [32].

3.6 Scattering Amplitudes of Quarks, Gluons and Bosons

An international team of physicists, in preparation for the Large Hadron Collider (LHC), is computing scattering amplitudes involving quarks, gluons and gauge vector bosons, in order to predict what results could be expected on the LHC. By default, these computations are performed using conventional double precision (64-bit IEEE) arithmetic. Then if a particular phase space point

PoS(ACAT08)014

is deemed numerically unstable, it is recomputed with double-double precision. These researchers expect that further optimization of the procedure for identifying unstable points may be required to arrive at an optimal compromise between numerical accuracy and speed of the code. Thus they plan to incorporate arbitrary precision arithmetic, using either the MPFUN90 or ARPREC packages, into these calculations. Their objective is to design a procedure where instead of using fixed double or quadruple precision for unstable points, the number of digits in the higher precision calculation is dynamically set according to the instability of the point [21].

In a related study, various checks of instabilities are employed, such as by comparing gluon amplitudes with known analytic values whenever possible. If a given point is deemed unstable by these tests, the researchers employ the QD package to re-evaluate the unstable points using higher precision (double-double or quad-double as needed). Because only a few points have to be re-computed to higher precision, they find that their average evaluation time is not significantly increased [16].

Two other recent examples of employing high-precision arithmetic in fundamental physics calculations of this type are [27] and [20].

3.7 Nonlinear Oscillator Theory

Quinn, Rand, and Strogatz recently described a nonlinear oscillator system by means of the formula

$$0 = \sum_{i=1}^{N} \left(2\sqrt{1 - s^2(1-2(i-1)/(N-1))^2} - \frac{1}{\sqrt{1 - s^2(1-2(i-1)/(N-1))^2}} \right).$$

They noted that for large N, $s \approx 1 - c/N$, where c = 0.6054436... These researchers asked the present authors and Richard Crandall to validate and extend this computation, and challenged us to identify this limit if it exists. By means of a Richardson extrapolation scheme, implemented on 64-CPUs of a highly parallel computer system, we computed (using the QD software)

$$c = 0.60544365719673274947892284244720747522089 96\ldots$$

This led to a proof that the limit c exists and is the root of a Hurwitz zeta function $\zeta(1/2, c/2) = 0$, where $\zeta(s,a) := \sum_{n \geq 0} 1/(n+a)^s$. As a bonus, we obtained some asymptotic terms [8].

3.8 Experimental Mathematics

High-precision computations have proven to be an essential tool for the emerging discipline of "experimental mathematics," namely the utilization of modern computing technology as an active agent of exploration in mathematical research [17][5]. One of the key techniques used here is the PSLQ integer relation detection algorithm [10]. An integer relation detection scheme is a numerical algorithm which, given an n-long vector (x_i) of real numbers (presented as a vector of high-precision floating-point values), attempts to recover the integer coefficients (a_i), not all zero, such that

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

PoS(ACAT08)014

(to available precision), or else determines that there are no such integers (a_i) such that the Euclidean norm $\sqrt{a_1^2 + a_2^2 + \cdots + a_n^2} < M$ for some bound M. The PSLQ algorithm operates by developing, iteration by iteration, an integer-valued matrix A which successively reduces the maximum absolute value of the entries of the vector $y = Ax$ (where x is the input vector mentioned above), until one of the entries of y is zero or within an "epsilon" of zero. 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 the algorithm must be performed using this precision level) or else the true relation will be lost in a sea of spurious numerical artifacts.

Perhaps the best-known application of PSLQ in experimental mathematics is the 1996 discovery of what is now known as the "BBP" formula for π:

$$\pi = \sum_{k=0}^{\infty} \frac{1}{16^k} \left(\frac{4}{8k+1} - \frac{2}{8k+4} - \frac{1}{8k+5} - \frac{1}{8k+6} \right).$$

This formula has the remarkable property that it permits one to calculate binary or hexadecimal digits beginning at the n-th digit, without needing to calculate any of the first $n-1$ digits, using a simple scheme that requires very little memory and no multiple-precision arithmetic software [4][17, pg. 135-143]. Since 1996, numerous other formulas of this type have been found, using the PSLQ-based computational approach, and then subsequently proven [17, pg. 147–149].

In an unexpected turn of events, it has been found that these computer-discovered formulas have implications for the age-old question of whether (and why) the digits of constants such as π and $\log 2$ are statistically random [11][17, pg. 163–174]. This same line of investigation has further led to a formal proof of normality (statistical randomness in a specific sense) for an uncountably infinite class of explicit real numbers. The simplest example of this class is the constant

$$\alpha_{2,3} = \sum_{n=1}^{\infty} \frac{1}{3^n 2^{3^n}},$$

which is provably 2-normal: every string of m binary digits appears, in the limit, with frequency 2^{-m} [12][17, pg. 174–178].

3.9 Ising Integrals

Several recent applications of high-precision computation have attempted to recognize definite integrals (typically arising in mathematical physics applications) using the methods of experimental mathematics. These computations have required the evaluation of integrals to very high precision, typically 100 to 1000 digits. In our studies, we have used either Gaussian quadrature (in cases where the function is well behaved in a closed interval) or the "tanh-sinh" quadrature scheme due to Takahasi and Mori [29] (in cases where the function has an infinite derivative or blow-up singularity at one or both endpoints). For many integrand functions, these schemes exhibit "quadratic" or "exponential" convergence – dividing the integration interval in half (or, equivalently, doubling the number of evaluation points) approximately doubles the number of correct digits in the result.

The tanh-sinh scheme is based on the observation, rooted in the *Euler-Maclaurin summation* formula, that for certain bell-shaped integrands (namely those where the function and all higher

PoS(ACAT08)014

derivatives rapidly approach zero at the endpoints of the interval), a simple block-function or trapezoidal approximation to the integral is remarkably accurate [2, 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 $h > 0$,

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

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 a recent study, the present authors together with Richard Crandall applied tanh-sinh quadrature, implemented using the ARPREC package, to study the following classes of integrals [7]. The D_n integrals arise in the Ising theory of mathematical physics, and the C_n have tight connections to quantum field theory.

$$\begin{aligned}
C_n &= \frac{4}{n!}\int_0^\infty \cdots \int_0^\infty \frac{1}{\left(\sum_{j=1}^n (u_j + 1/u_j)\right)^2}\frac{\mathrm{d}u_1}{u_1}\cdots\frac{\mathrm{d}u_n}{u_n} \\
D_n &= \frac{4}{n!}\int_0^\infty \cdots \int_0^\infty \frac{\prod_{i<j}\left(\frac{u_i-u_j}{u_i+u_j}\right)^2}{\left(\sum_{j=1}^n (u_j + 1/u_j)\right)^2}\frac{\mathrm{d}u_1}{u_1}\cdots\frac{\mathrm{d}u_n}{u_n} \\
E_n &= 2\int_0^1 \cdots \int_0^1 \left(\prod_{1\le j<k\le n}\frac{u_k-u_j}{u_k+u_j}\right)^2 \mathrm{d}t_2\,\mathrm{d}t_3\cdots dt_n,
\end{aligned}$$

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

Needless to say, evaluating these n-dimensional integrals to high precision presents a daunting computational challenge. Fortunately, in the first case, we were able to show that the C_n integrals can be written as one-dimensional integrals:

$$C_n \;=\; \frac{2^n}{n!}\int_0^\infty p K_0^n(p)\,\mathrm{d}p,$$

where K_0 is the *modified Bessel function* [1]. After computing C_n to 1000-digit accuracy for various n, we were able to identify the first few instances of C_n in terms of well-known constants, e.g.,

$$\begin{aligned}
C_3 &= \mathrm{L}_{-3}(2) \;=\; \sum_{n\ge 0}\left(\frac{1}{(3n+1)^2} - \frac{1}{(3n+2)^2}\right) \\
C_4 &= \frac{7}{12}\zeta(3),
\end{aligned}$$

PoS(ACAT08)014

where ζ denotes the Riemann zeta function. When we computed C_n for fairly large n, for instance

$$C_{1024} = 0.630473503374386796122040192710878904354587078712732 34\ldots,$$

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 numerical value can be identified as

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

where γ is Euler's constant. We later were able to prove this fact—this is merely the first term of an asymptotic expansion—and thus showed that the C_n integrals are fundamental in this context [7].

The integrals D_n and E_n are much more difficult to evaluate, since 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. In the case of D_5 and E_5, the resulting 3-D integrals are extremely complicated, but we were nonetheless able to numerically evaluate these to at least 240-digit precision on a highly parallel computer system. In this way, we produced the following evaluations, all of which except the last we subsequently were able to prove:

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

where Li denotes the polylogarithm function. In the case of D_2, D_3 and D_4, these are confirmations of known results. We tried but failed to recognize D_5 in terms of similar constants (the 500-digit numerical value is available if anyone wishes to try). The conjectured identity shown here for E_5 was confirmed to 240-digit accuracy, which is 180 digits beyond the level that could reasonably be ascribed to numerical round-off error; thus we are quite confident in this result even though we do not have a formal proof.

In a follow-on study [9], we examined the following generalization of the C_n integrals:

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

Here we made the initially surprising discovery—now proven in [18]—that there are linear relations in each of the rows of this array (considered as a doubly-infinite rectangular matrix), e.g.,

$$0 = C_{3,0}-84C_{3,2}+216C_{3,4}$$

PoS(ACAT08)014

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

In yet a more recent study, co-authored with physicists David Broadhurst and Larry Glasser [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). We also 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

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

where ${}_pF_q$ denotes the *generalized hypergeometric* function [1]. The corresponding odd values are $c_{3,1} = 3L_{-3}(2)/4$, $c_{3,3} = L_{-3}(2) - 2/3$, $c_{4,1} = 7\zeta(3)/8$ and $c_{4,3} = 7\zeta(3)/32 - 3/16$.

Integrals in the Bessel moment study were quite challenging to evaluate numerically. As one example, we sought to numerically verify the following identity that we had derived analytically:

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

where **K** denotes the elliptic integral of the first kind [1]. Note that this function has blow-up singularities on all four sides of the region of integration, with particularly troublesome singularities at $(\pi/2, -\pi/2)$ and $(-\pi/2, \pi/2)$ (see Figure 1). Nonetheless, after making some minor substitutions, we were able to evaluate (and confirm) this integral to 120-digit accuracy (using 240-digit working precision) in a run of 43 minutes on 1024 cores of the "Franklin" system at LBNL.

4. Conclusion

We have presented here a brief survey of the rapidly expanding applications of high-precision arithmetic in modern scientific computing. It is worth noting that all of these examples have arisen in the past ten years. Thus we may be witnessing the birth of a new era of scientific computing, in which the numerical precision required for a computation is as important to the program design as

PoS(ACAT08)014

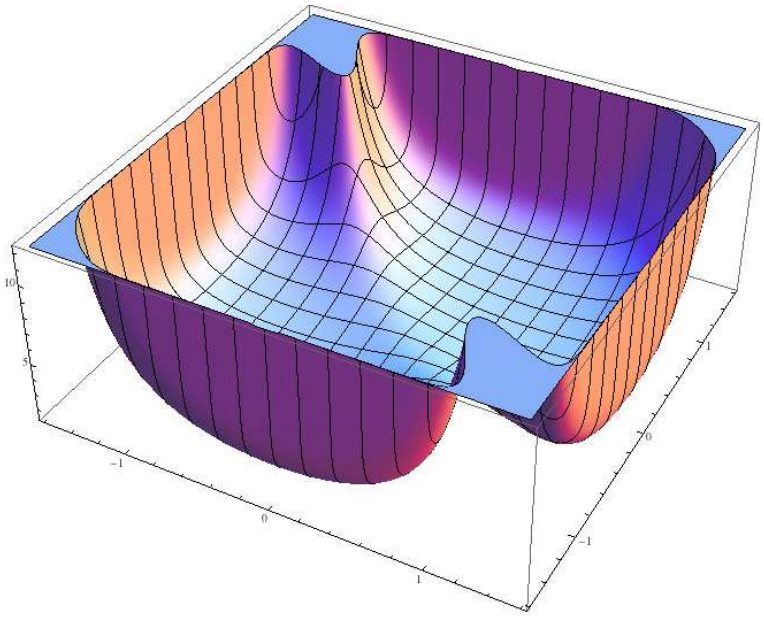

Figure 1: Plot of $c_{5,0}$ integrand function.

are the algorithms and data structures. We hope that our survey and analysis of these computations will be useful in this process.

Efforts to analyze integrals that arise in mathematical physics have underscored the need for significantly faster schemes to produce high-precision values of 2-D, 3-D and higher-dimensional integrals. Along this line, the "sparse grid" methodology has some promise [28][31]. Current research is aimed at evaluating such techniques for high-precision applications.

References

[1] M. Abramowitz and I. A. Stegun, ed., *Handbook of Mathematical Functions*, Dover, New York, 1972.

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

[3] D. H. Bailey, "Integer relation detection," *Comp. in Science and Engineering*, Jan-Feb., 2000, 24–28.

[4] D. H. Bailey, P. B. Borwein, and S. Plouffe, "On the rapid computation of various polylogarithmic constants," *Math. of Computation*, vol. 66, no. 218 (Apr 1997), 903–913.

[5] D. H. Bailey and J. M. Borwein, "Experimental mathematics: Examples, methods and implications," *Notices of the AMS*, vol. 52, no. 5 (May 2005), 502-514.

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

[7] D. H. Bailey, J. M. Borwein and R. E. Crandall, "Integrals of the Ising class," *J. Physics A: Math. and Gen.*, vol. 39 (2006), 12271-12302.

[8] D. H. Bailey, J. M. Borwein and R. E. Crandall, "Resolution of the Quinn-Rand-Strogatz constant of nonlinear physics," *Exp. Mathematics*, to appear, `http://crd.lbl.gov/~dhbailey/dhbpapers/QRS.pdf`.

PoS(ACAT08)014

[9] David H. Bailey, David Borwein, Jonathan M. Borwein and Richard Crandall, "Hypergeometric forms for Ising-class integrals," *Exp. Mathematics*, vol. 16 (2007), no. 3, 257-276.

[10] D. H. Bailey and D. Broadhurst, "Parallel integer relation detection: Techniques and applications," *Math. of Computation*, vol. 70, no. 236 (2000), 1719–1736.

[11] D. H. Bailey and R. E. Crandall, "On the random character of fundamental constant expansions," *Exp. Mathematics*, vol. 10, no. 2 (June 2001), 175–190.

[12] D. H. Bailey and R. E. Crandall, "Random generators and normal numbers," *Exp. Mathematics*, vol. 11, no. 4 (2004), 527–546.

[13] D. H. Bailey and A. M. Frolov, "Universal variational expansion for high-precision bound-state calculations in three-body systems. Applications to weakly-bound, adiabatic and two-shell cluster systems," *J. Physics B*, vol. 35, no. 20 (28 Oct 2002), 42870–4298.

[14] D. H. Bailey, X. S. Li and K. Jeyabalan, "A comparison of three high-precision quadrature schemes," *Exp. Mathematics*, vol. 14 (2005), no. 3, 317–329.

[15] E. Baron and P. Nugent, personal communication, Nov. 2004.

[16] C. F. Berger, Z. Bern, L. J. Dixon, F. Febres Cordero, D. Forde, H. Ita, D. A. Kosower and D. Maitre, "An automated implementation of on-shell methods for one-loop amplitudes," *Phys. Rev. D*, vol. 78 (2008), 036003, `http://arxiv.org/abs/0803.4180`.

[17] J. M. Borwein and D. H. Bailey, *Mathematics by Experiment: Plausible Reasoning in the 21st Century*.

[18] J. M. Borwein and B. Salvy, "A proof of a recursion for Bessel moments," *Exp. Mathematics*, vol. 17 (2008), 223–230.

[19] R. P. Brent and P. Zimmermann, *Modern Computer Arithmetic*, book manuscript, to appear, 2008.

[20] M. Czakon, "Tops from light quarks: Full mass dependence at two-Loops in QCD," *Phys. Lett. B*, vol. 664 (2008), 307, `http://arxiv.org/abs/0803.1400`.

[21] R. K. Ellis, W. T. Giele, Z. Kunszt, K. Melnikov and G. Zanderighi, "One-loop amplitudes for W+3 jet production in hadron collisions," manuscript, 15 Oct 2008, `http://arXiv.org/abs/0810.2762`.

[22] T. Ferris, *Coming of Age in the Milky Way*, HarperCollins, New York, 2003.

[23] A. M. Frolov and D. H. Bailey, "Highly accurate evaluation of the few-body auxiliary functions and four-body integrals," *J. Physics B*, vol. 36, no. 9 (14 May 2003), 1857–1867.

[24] P. H. Hauschildt and E. Baron, "The numerical solution of the expanding Stellar atmosphere problem," *J. Comp. and Applied Math.*, vol. 109 (1999), 41–63.

[25] Y. He and C. Ding, "Using accurate arithmetics to improve numerical reproducibility and stability in parallel applications," *J. Supercomputing*, vol. 18, no. 3 (Mar 2001), 259–277.

[26] G. Lake, T. Quinn and D. C. Richardson, "From Sir Isaac to the Sloan survey: Calculating the structure and chaos due to gravity in the universe," *Proc. of the 8th ACM-SIAM Symp. on Discrete Algorithms*, SIAM, Philadelphia, 1997, 1–10.

[27] G. Ossola, C. G. Papadopoulos and R. Pittau, "CutTools: a program implementing the OPP reduction method to compute one-loop amplitudes," *J. High-Energy Phys.*, vol. 0803 (2008), 042, `http://arxiv.org/abs/0711.3596`.

PoS(ACAT08)014

[28] S. Smolyak, "Quadrature and interpolation formulas for tensor products of certain classes of functions," *Soviet Math. Dokl.*, vol. 4 (1963), 240243.

[29] H. Takahasi and M. Mori, "Double exponential formulas for numerical integration," *Pub. RIMS*, Kyoto University, vol. 9 (1974), 721–741.

[30] Z.-C. Yan and G. W. F. Drake, "Bethe logarithm and QED shift for Lithium," *Phys. Rev. Letters*, vol. 81 (12 Sep 2003), 774–777.

[31] C. Zenger, "Sparse grids," in W. Hackbusch, ed., *Parallel Algorithms for Partial Differential Equations*, vol. 31 of *Notes on Numerical Fluid Mechanics*, Vieweg, 1991.

[32] T. Zhang, Z.-C. Yan and G. W. F. Drake, "QED corrections of $O(mc^2\alpha^7 \ln \alpha)$ to the fine structure splittings of Helium and He-Like ions," *Phys. Rev. Letters*, vol. 77, no. 26 (27 Jun 1994), 1715–1718.

PoS(ACAT08)014